MAINVIEW® SRM Enterprise Storage Automation User Guide

Version 7.1

March 28, 2002

Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED–RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b), or DFARS 227.7202, as amended from time to time. Send any contract notices to Contractor/Manufacturer:

BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827
USA

Contacting BMC Software

You can access the BMC Software Web site at **http://www.bmc.com**. From this Web site, you can obtain general information about the company, its products, special events, and career opportunities. For a complete list of all BMC Software offices and locations, go to **http://www.bmc.com/contact.html**.

USA and Canada

Address	BMC Software, Inc. 2101 CityWest Blvd. Houston TX 77042-2827
Telephone	713 918 8800 or 800 841 2031
Fax	713 918 8000

Outside USA and Canada

Telephone	(01) 713 918 8800
Fax	(01) 713 918 8000

Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see "Before Contacting BMC Software," below.

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, seven days a week by accessing the technical support Web site at **http://www.bmc.com/support.html**. From this site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support via Telephone or E-mail

In the USA and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the USA and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at **www.bmc.com/support.html**.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that a technical support analyst can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating-system and environment information
  - machine type
  - operating system type, version, and service pack or program temporary fix (PTF)
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or PTF
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software

Contents

List of Figures

List of Tables

About This Book

This book contains detailed information about MAINVIEW® Storage Resource Manager Enterprise Storage Automation by BMC Software. This guide is intended for storage administrators, systems programmers, data center support personnel, applications programmers, performance analysts, and anyone responsible for monitoring enterprise storage.

To use this book, you should be familiar with the following items:

- OS/390, job control language (JCL)
- Interactive System Productivity Facility (ISPF)
- A basic understanding of MAINVIEW is required to use Enterprise Storage Automation. If you have never used MAINVIEW products, read *Using MAINVIEW.*
- A basic understanding of MAINVIEW SRM is required to use Enterprise Storage Automation. If you have never used MAINVIEW SRM products, read the following chapters in the *MAINVIEW SRM User Guide and Reference*:
  - — the chapter pertaining to syntax to understand MAINVIEW SRM syntax and architecture
  - — the chapter pertaining to SVOS to understand how MAINVIEW SRM components are started and stopped and how they communicate
  - — the chapter pertaining to functions to understand how MAINVIEW SRM component functions control the runtime services of MAINVIEW SRM

- A basic understanding of MAINVIEW AutoOPERATOR is required to use Enterprise Storage Automation. If you have never used MAINVIEW AutoOPERATOR, read the AutoOPERATOR documentation, starting with *MAINVIEW AutoOPERATOR Basic Automation Guide*.

How This Book Is Organized

This book is organized as follows. In addition, an index and glossary appear at the end of the book.

Chapter/Appendix	Description
Chapter 1, “Introduction”	provides a brief overview of Enterprise Storage Automation
Chapter 2, “Basic Event Generation”	describes how to define user event solutions, activate and inactivate system event solution modes, and override default system event solution definition values
Chapter 3, “Automation Overview”	describes how Enterprise Storage Automation responds to events with a solution for the condition using automation
Chapter 4, “Using AUTO Functions”	defines the functions used for automation
Chapter 5, “Using AUTO Commands”	defines the console commands used to initiate AUTO function processing
Chapter 6, “Implementing and Customizing Solutions”	describes distributed solutions that you can modify and use to automate your storage tasks
Appendix A, “System Event Definitions”	provides an explanation of system-generated event definitions
Appendix B, “REXX EXECs”	provides a detailed description of the REXX EXECs used for automation

Related Documentation

BMC Software products are supported by several types of documentation:

- online and printed books
- online Help
- release notes and other notices

In addition to this book and the online Help, you can find useful information in the publications listed in the following table. As “Online and Printed Books” on page xvi explains, these publications are available on request from BMC Software.

Category	Document	Description
MAINVIEW common documents	*OS/390 and z/OS Installer Guide* *MAINVIEW Installation Requirements Guide* *MAINVIEW Common Customization Guide* *Using MAINVEW* *MAINVIEW Administration Guide* *Implementing Security for MAINVIEW*	provide instructions for installing, configuring, using, and administering MAINVIEW
MAINVIEW SRM customization documents	*MAINVIEW SRM Customization Guide*	provides instructions for configuring and customizing MAINVIEW SRM for OS/390 including Enterprise Storage Automation
core documents	*MAINVIEW SRM User Guide and Reference*	provides information common to all MAINVIEW SRM products and high-level navigation
	MAINVIEW SRM Reference Summary	provides information about global system parameters, filter and rule list parameters, and functions for all MAINVIEW SRM products
messages	*MAINVIEW SRM Messages*	provides hardcopy of MAINVIEW SRM messages that are also available online
other MAINVIEW SRM product documents	MAINVIEW SRM *EasyHSM User Guide and Reference* MAINVIEW SRM *EasyPOOL User Guide and Reference* MAINVIEW SRM *EasySMS User Guide and Reference* MAINVIEW SRM *SG-Control User Guide and Reference* MAINVIEW SRM *StopX37/II User Guide and Reference* MAINVIEW SRM *StorageGUARD User Guide and Reference*	contain information you need to generate events through MAINVIEW SRM filter list and rule list technology
other BMC required products	*AutoOPERATOR Advanced Automation Guide for CLIST EXECs* *AutoOPERATOR Advanced Automation Guide for REXX EXECs* *AutoOPERATOR Basic Automation Guide* *AutoOPERATOR Customization Guide* *AutoOPERATOR Options User Guide* *AutoOPERATOR Reference Summary*	contain information you need to install and use the AutoOPERATOR component of MAINVIEW SRM
supplemental documents	release notes, flashes, technical bulletins	provides additional information about the product

Online and Printed Books

The books that accompany BMC Software products are available in online format and printed format. If you are a Windows or Unix user, you can view online books with Acrobat Reader from Adobe Systems. The reader is provided at no cost, as explained in “To Access Online Books.” You can also obtain additional printed books from BMC Software, as explained in “To Request Additional Printed Books.”

To Access Online Books

Online books are formatted as Portable Document Format (PDF) files. You can view them, print them, or copy them to your computer by using Acrobat Reader 3.0 or later.You can access online books from the documentation compact disc (CD) that accompanies your product or from the World Wide Web.

In some cases, installation of Acrobat Reader and downloading the online books is an optional part of the product-installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at **http://www.adobe.com**.

To view any online book that BMC Software offers, visit the support page of the BMC Software Web site at **http://www.bmc.com/support.html**. Select a product to access the related documentation.

To Request Additional Printed Books

BMC Software provides printed books with your product order. To request additional books, go to **http://www.bmc.com/support.html.**

Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information such as

- updates to the installation instructions
- last-minute product information

In addition, BMC Software sometimes provides updated product information between releases (in the form of a flash or a technical bulletin, for example). The latest versions of the release notes and other notices are available on the Web at **http://www.bmc.com/support.html**.

Conventions

This section provides examples of the conventions used in this book and explains how to read ISPF panel-flow diagrams and syntax statements.

General Conventions

This book uses the following general conventions:

Item	Example
information that you are instructed to type	Type **SEARCH DB** in the designated field.
specific (standard) keyboard key names	Press **Enter**.
field names, text on a panel	Type the appropriate entry in the **Command** field.
directories, file names, Web addresses	The BMC Software home page is at **www.bmc.com**.
nonspecific key names, option names	Use the HELP function key. KEEPDICTIONARY option
calls, commands, control statements, keywords, parameters, reserved words	Use the SEARCH command to find a particular object. The product generates the SQL TABLE statement next.

code examples, syntax statements, system messages, screen text	`//STEPLIB DD`
	`The table` *`table_name`* `is not available.`

emphasized words, new terms, variables	The instructions that you give to the software are called *commands*. In this message, the variable *file_name* represents the file that caused the error.
single-step procedures	» To enable incremental backups, type **y** and press **Enter** at the next prompt.

This book uses the following types of special text:

Note: Notes contain important information that you should consider.

Warning! Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

Tip: Tips contain useful information that may improve product performance or that may make procedures easier to follow.

Syntax Statements

Syntax statements appear in Courier. The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2|KEYWORD3] KEYWORD4={YES|NO}
       file_name...
```

The following table explains conventions for syntax statements and provides examples:

Item	Example
Items in italic type represent variables that you must replace with a name or value. Use an underscore for variables with more than one word.	`dtsbackup` *control_directory*
Brackets indicate a group of options. You can choose at least one of the items in the group, but none of them is required. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.	[*table_name, column_name, field*]
Braces enclose a list of required items. You must enter at least one of the items. Do not type the braces when you enter the item.	{*DBD_name* | *table_name*}
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either *commit* or *cancel*.	`{commit | cancel}`
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	*column_name* . . .

Chapter 1 Introduction

MAINVIEW SRM Enterprise Storage Automation delivers powerful event generation and storage automation technology across the storage enterprise. With Enterprise Storage Automation, storage related events can be detected anywhere throughout the MAINVIEW SRM suite of products. Automated solutions to perform pool, volume, application, or even data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests.

With the Enterprise Storage Automation AUTO functions, monitoring and automated storage management of pools, groups, applications, and volumes can be performed, invoking distributed or user-created automation solutions when utilization or other thresholds are exceeded. These solutions can be initiated in a tiered or multi-level approach, where solutions can be implemented one after another, with each solution taking a more drastic action, until the condition is resolved.

The following topics provide an overview of Enterprise Storage Automation:

Terms

Review the following terms to familiarize yourself with basic event identification and automation process terminology:

AUTO commands
refers to the Enterprise Storage Automation console commands that are used to initiate automation: AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL

AUTO functions
refers to the Enterprise Storage Automation functions that are used to define and activate storage automation: AUTOAPPL, AUTODS, AUTOPOOL, and AUTOVOL

event a user-defined message sent to AutoOPERATOR by MAINVIEW SRM

User-defined storage occurrences can cause events to be sent to AutoOPERATOR, where actions can be initiated in response to the storage condition.

fire the term used to indicate that AutoOPERATOR has initiated an action in response to an event being received.

In AutoOPERATOR, when a rule selection criteria matches an incoming event and *fires*, the user-specified automation actions are performed.

FLST/RLST
short for filter list and rule list

The filter list refers to SMFLST*xx* parameters that control selection of resources for a function. The rule list refers to SMRLST*xx* parameters that include a SET statement and one or more INC/EXC statements. For further information, see the *MAINVIEW SRM User Guide and Reference*.

MAINVIEW AutoOPERATOR
the MAINVIEW product that uses tools, techniques, and facilities to automate routine operator tasks

AutoOPERATOR is used in conjunction with Enterprise Storage Automation to automate the solutions to storage problems.

Multi-Level Automation (MLA)

the user-defined multi-step process in Enterprise Storage Automation that implements solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved

pool general term used *in the context of this book* to mean either an SMS Storage Group, a MAINVIEW SRM-defined pool, or a MAINVIEW SRM-defined SMS Pool

SET result group

a group of records that matched the same AUTO function FLST/RLST SET statement

The AUTO function FLST/RLST processes slightly differently than most other MAINVIEW SRM functions. The AUTO functions FLST/RLST create a group of records according to the INC/EXC statements of the SET. The SET statement keywords then determine how to process the group of records called the *SET result group*. This allows sorting of the result group and actions implemented in sort order.

Skeleton Tailoring Facility

a facility in AutoOPERATOR that allows skeleton JCL to be used during job submission.

Skeleton JCL can contain variables within the JCL statements, to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility is similar to the TSO skeleton tailoring facility in function. This facility is described the the IMFEXEC TAILOR command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

solutions

an action that is taken to affect a condition, hopefully resolving it

Solutions are defined in AUTO function FLST/RLST SET statements and determine the automated action to take and the resources to take it on. Solutions are requested through the AUTO console commands.

A solution either submits a job to manipulate resources or generates events that can then further analyze the condition.

stem variables

a REXX facility, supported in AutoOPERATOR REXX EXECs and Skeleton Tailoring Facility, where variable names end with a period followed by a number, such as &POOL.1

This allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array, for example &POOL.0 = 5 would indicate variables &POOL.1 through &POOL.5 exist.

See the Glossary at the end of this book for definitions of other terms you may find in this book.

Why Use Enterprise Storage Automation?

With Enterprise Storage Automation, you can

- automate pool, group, and application space management functions in addition to SMS and HSM processing
- create your own solutions to automate whatever volume or data set manipulation you perform when you absolutely have to have space, right now
- use the AutoOPERATOR rules-based processor to automate actions in response to any storage-related condition noted in any MAINVIEW SRM function
- implement and customize distributed solutions
- view storage event statistics and automation activity
- implement HSM automation on held or waiting resources, duplicate requests, WTOR replies, message suppression, and hung sessions.

How Events Work

Enterprise Storage Automation can externalize any user-defined storage condition noted in FLST/RLST members of MAINVIEW SRM functions. When the condition is identified, an event, in the form of a user-defined message, can be generated and sent to AutoOPERATOR. An event message does not appear in any joblog or on the MAINVIEW SRM console; it is sent directly to AutoOPERATOR. The user-defined text of the event message is specified in an event definitions member. Figure 1-1 shows the relationships of the various members that come together to generate an event.

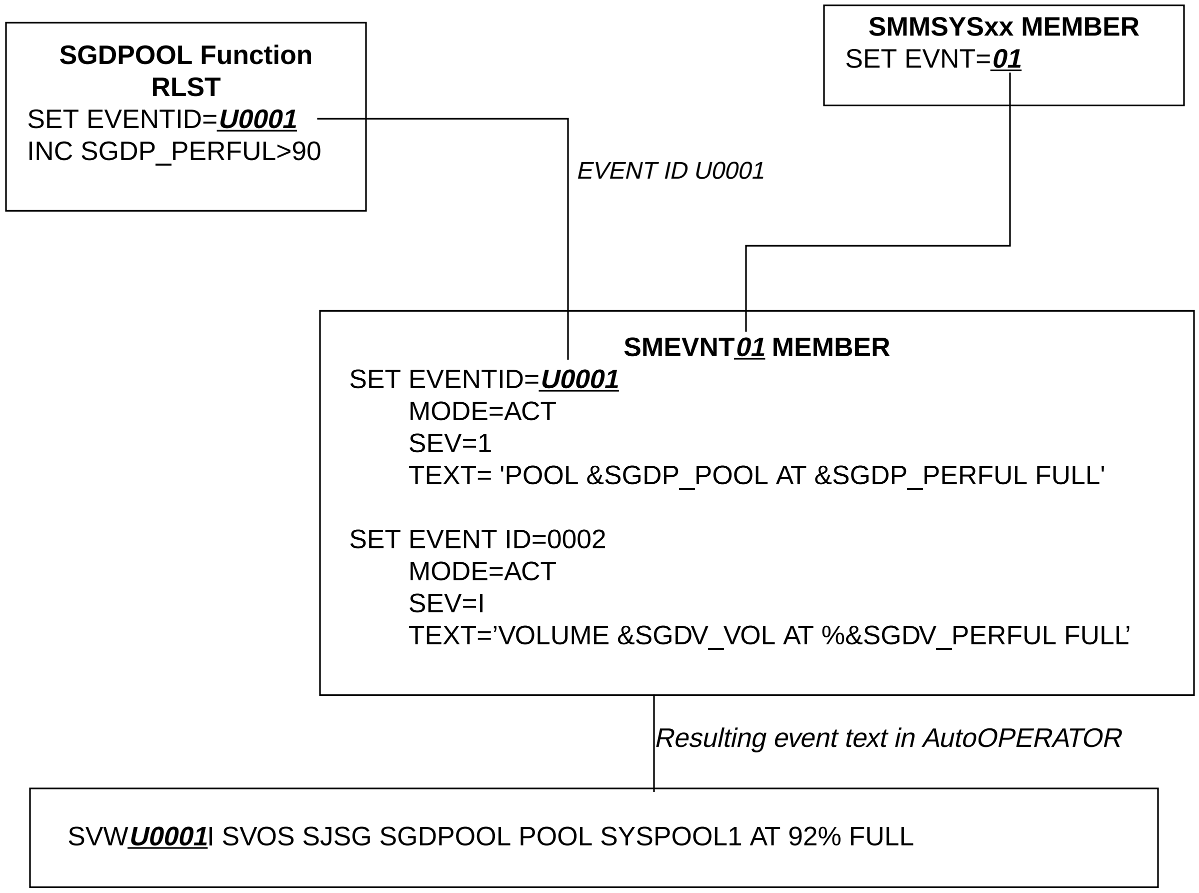
The EVENTID= keyword defines the identification of the event. In the SMEVNT*xx* parmlib member, you define the entire event definition, including modifiable text with data substitution.

The SMEVNT*xx* member is also used to activate internal system events generated automatically by MAINVIEW SRM.

You must create an entry in SMEVNT*xx* for each user event that you create and each system event that you activate.

The SMEVNT member suffix is specified in the SMMSYS*xx* member by way of the EVNT= keyword. SMMSYS*xx* is the MAINVIEW SRM system configuration member and contains suffixes of most other parmlib members. For further information, see Chapter 2, “Basic Event Generation.”

Figure 1-1 Event Members



SGDPOOL Function RLST

```
SET EVENTID=U0001
INC SGDP_PERFUL>90
```

SMMSYSxx MEMBER

```
SET EVNT=01
```

SMEVNT01 MEMBER

```
SET EVENTID=U0001
        MODE=ACT
        SEV=1
        TEXT= 'POOL &SGDP_POOL AT &SGDP_PERFUL FULL'

SET EVENT ID=0002
        MODE=ACT
        SEV=I
        TEXT=’VOLUME &SGDV_VOL AT %&SGDV_PERFUL FULL’
```

```
SVWU0001I SVOS SJSG SGDPOOL POOL SYSPOOL1 AT 92% FULL
```

How Automation Works

Enterprise Storage Automation provides the ability to create automated solutions to perform pool, volume, application, or data set-level storage management functions. These user-defined solutions, or any of a number of distributed solutions, can be initiated in response to almost any storage condition or by user request through the use of the AUTO console commands.

Solutions are defined in FLST/RLST statements of the AUTO functions. Like most MAINVIEW SRM functions, these statements first select the resources or elements to be acted on through INC/EXC statements and then specify the actions to be taken on the selected elements through SET statement keywords. With the AUTO functions, the INC/EXC keywords select the pools, volumes, applications, or data sets to be acted on and the SET keywords control event generation and job submission actions on the selected elements. These actions can be used to either

- externalize the selection of the resource or element by issuing an event

  OR

- take action by tailoring skeleton JCL to include statements for each selected element and submitting a batch job to perform a space management function on those elements

  Skeleton tailoring JCL can invoke any utility software. The skeleton tailoring facility drives other vendors' disk management systems, standard IBM utilities, and in-house utilities. Skeleton Tailoring output is a submitted batch job JCL. For example, a submitted batch job could be used to migrate selected data sets, perform first-level space management, defrag a volume(s), reorg data set(s), or any other pool, volume, or data set function you can invoke in a batch job.

The AUTO functions and their solutions are requested through the AUTO console commands. Unlike most MAINVIEW SRM functions, which correspond to a point during a process, such as allocation, the AUTO functions process only by console request. Console commands can be issued

- **manually**, to initiate solutions for those ad hoc requests where you have to make some space

  OR

- **automatically**, through AutoOPERATOR rules or through any other mechanism that allows the issuance of console commands (MAINVIEW Alarm, MAINVIEW API within AutoOPERATOR, AutoOPERATOR REXX EXECs, and so on)

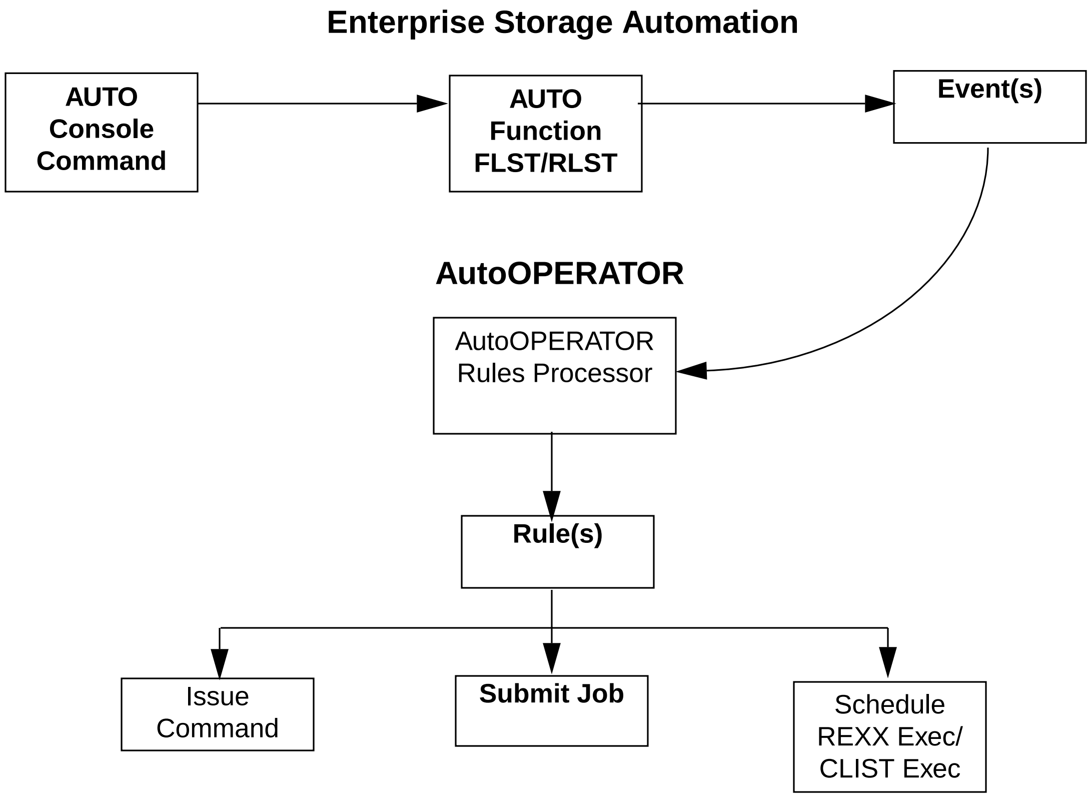
In conjunction with AutoOPERATOR, Enterprise Storage Automation solutions can be used to provide completely automated storage management of pools, volumes, and applications. The AUTO functions can be used to perform the initial automation step of identifying a storage condition when the condition relates to pools, volumes, or applications. AutoOPERATOR, and its rules processor, can respond to events with a number of possible actions, including issuing an OS/390 command. This command can initiate an Enterprise Storage Automation solution by making use of the AUTO commands. Thus, in summary, when a condition is noted and an event is generated, an AutoOPERATOR rule is used to initiate a solution for the condition, issuing the AUTO console command.

Solutions can be initiated in a tiered approach, where each solution is initiated in order until the original condition is resolved. This is called multi-level automation.

Whether implementing a distributed solution or creating your own solution, you need a thorough understanding of AutoOPERATOR, Enterprise Storage Automation, and MAINVIEW SRM function FLST/RLST statements.

Figure 1-2 provides you with a generic process flow and shows you how Enterprise Storage Automation works with AutoOPERATOR to automate storage solutions. See Chapter 3, “Automation Overview” for a detailed description of the automation process flow.

Figure 1-2 Automation Process Flow



Controlling Enterprise Storage Automation

Enterprise Storage Automation (SVESA) can be started any time after SVOS is started. AutoOPERATOR can be started before or after SVESA and can be cycled while SVESA is active. Use the following SVOS console command to start Enterprise Storage Automation:

/S SVESA	using subchar
F SVOS,S SVESA	not using subchar

Use the following SVOS console command to stop Enterprise Storage Automation:

```
/P SVESA
```

Stopping and restarting Enterprise Storage Automation can be used to resolve most automation-related problems. However, when SVESA is stopped and restarted, the AUTO view will be empty. Any batch jobs already submitted by automation will not be affected and will continue to run to completion. At completion, these jobs will still send a job-end notification to SVOS (as described in “Job Submission and Skeleton Tailoring” on page 4-29); however, the notification will be ignored. Any multi-level processing is terminated. The event counts on the EVENTS view will also be reset to zero.

Refreshing User and System Events

You can refresh a user events by using the refresh line command. You can also use an SVOS console command to refresh events. System events cannot be refreshed; they are defined internally and do not change.

The following command rebuilds all SVOS system definition members, including the SMEVNT*xx* member.

```
/SVOS R,EVNT=xx
```

The variable *xx* represents the suffix of the event member.

The following command rebuilds Enterprise Storage Automation and refreshes parameters for all events specified in SMMSYS*xx*, including SMEVNT*xx*:

```
/ SVOS R,SYS=xx
```

The variable *xx* represents the suffix of the SMMSYS*xx* member.

For more information about SVOS, see the *MAINVIEW SRM User Guide and Reference*.

Requirements and Restrictions

Before you begin, review the following requirements and restrictions for using Enterprise Storage Automation:

- For volume automation to occur, at either the pool, volume, or data set level, the volume must be online. Any offline volume condition is ignored and no information on the volume is collected.

- SG-Control must be active to perform any application automation. AUTOAPPL requests require SG-Control to be active.

- SMS must be installed to enable use of the following fields in the AUTOVOL FLST/RLST INC/EXC statements.

fragmentation index	AV_FRAGI
VTOC status	AV_VTOCI
volume SMS Status	AV_SMSI
free DSCBs	AV_FREED
free VIRs	AV_FREEV

- Current utilization of a pool is calculated by dividing the total space allocated in the pool by the total capacity of the pool. All calculations are performed in bytes in order to provide device independence.

Chapter 2 Basic Event Generation

This chapter contains the information that you need to

- define filter and rule list statements to note a condition and the EVENTID= keyword to issue an event
- create user event definitions in the SMEVENT*xx* member
- establish AutoOPERATOR subsystems in the SMMSYS*xx* member that communicate with the MAINVIEW SRM

The following topics provide an explanation of how to identify storage conditions, define and activate events, and establish communication with AutoOPERATOR subsystems:

Identifying Storage Conditions with FLST/RLST

Thresholds and conditions that cause events to be noted are defined by SET statement and INC/EXC statement parameters in FLST*xx* and RLST*xx* MAINVIEW SRM parmlib members. In FLST*xx* and RLST*xx* members

- the event ID is specified on SET statement parameters
- the thresholds and definitions that identify the storage condition are specified on INC/EXC statement parameters that follow the SET statement

SET Statement Parameters

The SET statement parameter in the following table is used to identify user events:

Parameter	Description
EVENTID	identifies a user event defined in an SMEVNT*xx* member

INC/EXC Statement Parameters

With the exception of the USERVARS function and the AUTO functions, filter list or rule list member INC/EXC statement parameters may be used to generate a user event for any installed MAINVIEW SRM component function. (AUTO functions use the ACT_EVENTID parameter, which is explained in Chapter 4, “Using AUTO Functions.”)

Note: The MODE parameter in an FLST member affects only function processing. It does not affect event generation. For example, if you specify MODE=INACT and EVENTID=U00001 in an FLST member, processing of the function will be bypassed but event U00001 will be generated. To turn an event off, you must specify MODE=INACTIVE in SMEVNT*xx* member for the U0001 event definition.

The *MAINVIEW SRM Reference Summary* contains a quick reference guide for the MAINVIEW SRM functions and components that support the functions. For more information, see the component user guides.

Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to the user event in SMEVNT*xx*.

Syntax: EVENTID=*xxxxx*

The variable *xxxxx* represents the 5-character string specified on the EVENTID parameter in SMEVENT*xx*. The value I is reserved for system events and must not be used in the first position of a user-defined EVENTID. With the exception of the letter I, you can use any 5 characters or numbers, in any position of the EVENTID for a user-defined event. You may want to create a naming convention for events that you define.

Events distributed as part of the solutions will begin with AP*xxx*, AA*xxx*, AV*xxx*, and AD*xxx*. You are not restricted from using these event IDs, but it is not recommended by BMC Software.

Required: Yes

Default: None

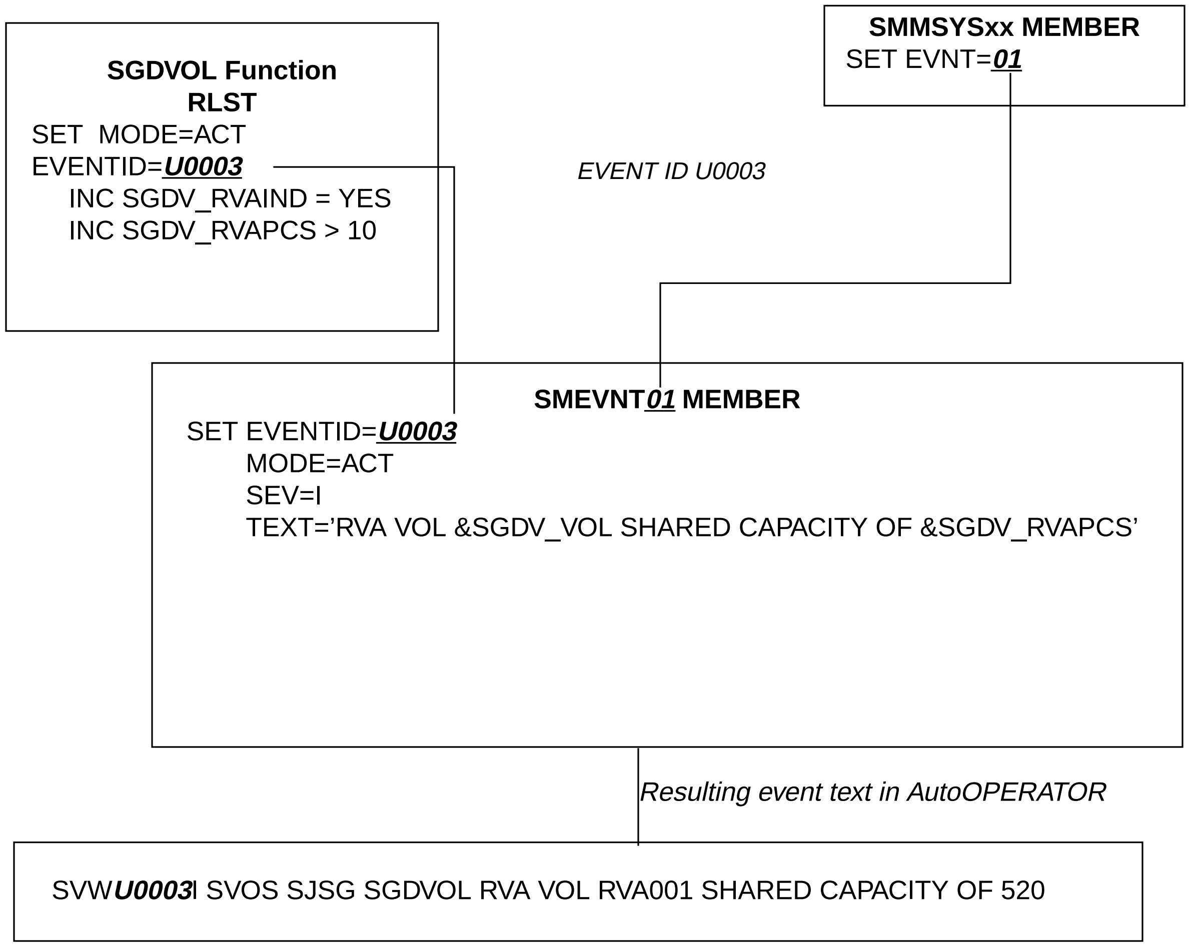
Example

The parameters in this example are used in the SGDVOL function to generate event U0003 whenever an RVA volume has a shared physical capacity greater than 1 MB (the field is in one-tenth MB values).

```
SET  MODE=ACT  EVENTID=U0003
     INC SGDV_RVAIND = YES SGDV_RVAPCS > 10
```

Figure 2-1 shows the relationship of the example members.

Figure 2-1 Event Example



SGDVOL Function RLST

```
SET  MODE=ACT
EVENTID=U0003
    INC SGDV_RVAIND = YES
    INC SGDV_RVAPCS > 10
```

SMMSYSxx MEMBER

```
SET EVNT=01
```

EVENT ID U0003

SMEVNT01 MEMBER

```
SET EVENTID=U0003
      MODE=ACT
      SEV=I
      TEXT='RVA VOL &SGDV_VOL SHARED CAPACITY OF &SGDV_RVAPCS'
```

Resulting event text in AutoOPERATOR

```
SVWU0003I SVOS SJSG SGDVOL RVA VOL RVA001 SHARED CAPACITY OF 520
```

Defining the Event Using SMEVNTxx

An event is a message resulting from an exceeded threshold or condition for a resource. Thresholds and conditions are defined in FLST/RLST members for MAINVIEW SRM functions.

Text of an event message can include any data available in the FLST/RLST for the function generating the event, that is, a function that includes an EVENTID statement.

The EVENTID specified in the function FLST/RLST must then be defined in the system event member (SMEVNT*xx*). You must create an entry in SMEVNT*xx* for each user event that you want to establish.

The SET statement parameters in the following table are used to identify and define events in the SMEVNT*xx* member.

Parameter	Description
EVENTID	identifies the event
MODE	sets the event to active or inactive
SEV	indicates the urgency of the event
TEXT	specifies the text of the event message
OVERRIDE	specifies that default system event parameters are to be replaced

Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to the user event in SMEVNT*xx*.

Syntax: EVENTID=*xxxxx*

The variable *xxxxx* represents the 5-character string specified on the EVENTID parameter in SMEVENT*xx*. The value I is reserved for system events and must not be used in the first position of a user-defined EVENTID. With the exception of the letter I, you can use any 5 characters or numbers, in any position of the EVENTID for a user-defined event. You may want to create a naming convention for events that you define.

Events distributed as part of the solutions will begin with AP*xxx*, AA*xxx*, AV*xxx*, and AD*xxx*. You are not restricted from using these event IDs, but it is not recommended by BMC Software.

Required: Yes

Default: None

MODE=

Purpose: Sets an event to active or inactive to turn event generation off or on.

If the event mode is inactive, event generation will be bypassed when the function that generates it is processed.

Syntax: MODE=ACTIVE/INACTIVE

Required: Yes

Default: For user events there is no default, the MODE= keyword must be specified. For internal system events the default is inactive. If you want to activate a system event, you must change the value on this parameter to MODE=ACTIVE.

SEV=

Purpose: Indicates the urgency of an event.

The severity indicator is appended to the end of SVW*xxxxx* to form a 9-character header for an event message.

Syntax: SEV=*x*

The variable *x* represents a single alphabetic or numeric character. It is recommended that you use one of the following characters:

— **I**–(informational messages)
— **W**–(warning messages)
— **E**–(error messages)
— **S**–(serious error messages)

Required: No

Default: None

TEXT=

Purpose: Specifies the text of the event message

Syntax: TEXT='*xxxxx...'*

The variable *xxxxx* represents the text enclosed in single quotation marks (') and can contain variables from the function generating the event. The total length of the text can be a maximum of 218 bytes once the variables are expanded. If the text is greater than 218 bytes after variable expansion, it is truncated word by word until it is 218 bytes or less.

Variables used on the TEXT= parameter must be based on INC/EXC statement parameters for functions that generate events. A text variable consists of an ampersand (&) followed by an INC/EXC statement parameter name valid for the function that generates the event. The ampersand does not have to proceeded by a space; however, there must be a space following the variable name.

When the event is generated, the value of the parameter is passed to the event and replaces the parameter name in the text. If you specify an invalid parameter name, Enterprise Storage Automation cannot replace the variable in the text with a data value. When a text variable error is encountered, one of the following messages rather than a data value is displayed in the text:

&*variable*(INVALID FOR FUNCTION)

indicates the parameter name used in the variable is not a valid parameter for the function that generates the event

&*variable*(VARIABLE NAME INVALID)

indicates the parameter name used in the variable contains more than 11 characters (parameter names are 11 characters or less) or that the variable specified is not a valid MAINVIEW SRM parameter

&*variable*(TEXT INVALID)

indicates that either an ampersand(&) was specified without a parameter name in the variable or an error occurred when the parser was attempting to locate the data value for the parameter specified

&*variable*(NOT INC/EXC PARAMETER)

indicates the parameter name used in the variable is not a valid INC/EXC statement parameter for the function attempting to generate the event and is probably a SET statement parameter

Note: Parameters used as text variables are restricted to INC/EXC statement parameters. You may not use SET statement parameter names as variables.

To continue a line of text to the next line, place a nonblank character in column 72 of the line to be continued. The first character in the next line is appended to the last character in the previous line. If you need a blank space to appear in the text following the character in column 72, place a quotation mark (') in the first position of the new line and a space after the quotation mark.

Required: Yes

Default: None

Example

The parameters in this example are for the AUTOPOOL function and display information from the AUTOPOOL record.

```
SET EVENTID=AP092
    SEV=W
    MODE=ACT
    TEXT='&AUTOLEV &AP_POOL %&AP_PERFL FULL CONTAINS &AP_VOLC VOLUMES'
```

The resulting event message for this example is:

```
SVWAP092W AUTOPOOL MVSRM SJSG AUTOLEV1 WRK001 95%
FULL CONTAINS 23 VOLUMES
```

OVERRIDE=

Purpose: allows you to replace default values for system events.

Syntax: OVERRIDE=Y/N

When OVERRIDE=Y is specified in an entry, the values you specify on the other parameters in the entry replace the system event default values. When OVERRIDE=Y is specified, the value in EVENTID= must be a MAINVIEW SRM-defined internal system event. These events are documented in Appendix A, “System Event Definitions.”

The MODE= and the TEXT= keywords can be used to override the corresponding information in the system event. When using TEXT= to override a system event, variable substitution is not available.

Required: No

Default: None

The following SET statement examples show how event definition parameters can be used:

Example

The parameters in this example override system event I1001 to a mode of active.

```
SET  EVENTID=I1001
     OVERRIDE=Y
     MODE=ACT
```

Event Message Format

The parameters you use in the SMEVNT*xx* member formulate most of the event message text. The message number and first three fields are automatically generated for every event. The user’s text starts after the fourth word in the following format:

SVW*xxxxxY* {*func_name* | SYSTEM} *job_name sys_id text*

where *xxxxx* is the *event_id* used to generate the event and *Y* is the severity code from the SMEVNTxx definition. The value SVW is the prefix for user and system events.

The value I is reserved for system events and must not be used in the first position of a user-defined *event_id*. With the exception of the letter I, you can use any 5 characters or numbers, in any position of the *event_id* for a user-defined event. You may want to create a naming convention for events that you define.

Events distributed as part of the solutions will begin with AP*xxx*, AA*xxx*, AV*xxx*, and AD*xxx*. You are not restricted from using these event IDs, but it is not recommended by BMC Software.

Example

For a *user* event with an *event_id* of AP001, the event/message number generated is

SVWAP001

For an internal system event with an *event_id* of I1001 (letter I, one, zero, zero, one), the event message number generated is:

SVWI1001

The variable *sev_ code* represents a user-specified value on a SEV parameter if the event is a user event. If the event is a system event, the code is either supplied by the system or is overridden with a user-specified value.

The variable *func_name* represents the 8-byte name of the function that generated the event if the event is a user event. If the event is a system event, the value SYSTEM is displayed.

The variable *job_name* represents the 8-byte name of the SVOS started task.

The variable *sys_id* represents the 8-byte identifier of the OS/390 system that was running when the event was generated.

The variable *text* represents the 218-byte message text defined by the user on a TEXT parameter if the event is a user event. If the event is a system event, the message is either supplied by the system or is overridden with a user-specified message.

Establishing System Parameters for Events

To activate user and system events referenced in the SMEVNT*xx* member, you must identify the SMEVNT*xx* member in the SMMSYS*xx* parmlib member using the EVNT parameter.

In order for Enterprise Storage Automation to link to AutoOPERATOR, the connection must be defined in the SMMSYS*xx* parmlib member using the AOO_SUBSYS parameter.

These parameters are described in detail below.

Parameter	Description
EVNT	activates the event definition member to be used for the configuration of MAINVIEW SRM that is running
AOO_SUBSYS	connects Enterprise Storage Automation to the AutoOPERATOR subsytem(s) to receive events for viewing on an AutoOPERATOR console

Parameter Explanations

EVNT=

Purpose: Specifies the suffix you assign to the name of the SMEVNT*xx* event definition member; for more information, see “Defining the Event Using SMEVNTxx” on page 2-5.

This value can be overridden after SVOS has started using the SVOS REFRESH command. To determine the currently active SMEVNT*xx* member, use the ADMEMA view or look in the SVOS joblog for the last occurrence of message SVM0731I.

Syntax: EVNT=*xx*

The variable *xx* represents the 2-character suffix of the SMEVNT*xx* member name.

Required: no

Default: 00

AOO_SUBSYS=

Purpose: Specifies the AutoOPERATOR subsytems that are to receive and submit jobs on behalf of MAINVIEW SRM requests. You may specify up to three AutoOPERATOR subsystems. No other automation product subsystem can be specified here.

Each event issued by Enterprise Storage Automation is sent to each of the AutoOPERATOR subsystems specified.

AutoOPERATOR is used by Enterprise Storage Automation to cause job submission when the ACT_JOB keyword of the AUTO functions FLST/RLST is used. For this processing the FIRST subsystem specified here is used to submit all jobs. This AutoOPERATOR subsystem must be at least V6.1 of AutoOPERATOR with PTF BPO5425 applied.

During Enterprise Storage Automation startup, each AutoOPERATOR in this list is connected. AutoOPERATOR may be cycled any time while Enterprise Storage Automation is active. Enterprise Storage Automation will notice this situation and suspend event generation to that AutoOPERATOR. If the first AutoOPERATOR in the list is stopped, job submission is also suspended. A message is issued to the SVOS joblog to indicate this situation. When the AutoOPERATOR is restarted, Enterprise Storage Automation will automatically reconnect with the AutoOPERATOR. Again, a message is issued to the SVOS joblog to indicate this.

Syntax: AOO_SUBSYS=(*xxxx*,*xxxx*,*xxxx*)

The variables *xxxx* represent the 2–4-character subsystem names that correspond to the values specified on the SS parameter in the AutoOPERATOR JCL.

Required: Yes

Default: none

Where to Go From Here

Thus far, you have seen:

- how to note a situation using the FLST/RLST filter and selection statements
- how to generate an event when the situation is noted using the SET keyword EVENTID=
- how to activate and define the event text for the event in SMEVNT*xx* and how to activate MAINVIEW SRM internal system events using this member

So, now you know how to externalize storage-related conditions to AutoOPERATOR using event generation. Some events stop at this point with having generated a message to alert someone of a problem. In other cases, you may want to take it a step further and automate a solution or solutions to resolve a problem without operator intervention. The automation process builds on the information in this chapter. See Chapter 3, “Automation Overview” for basic methods of automating storage solutions; see “Multi-Level Automation (MLA) Processing” on page 4-35 for even more advanced methods of automating storage solutions.

Chapter 3 Automation Overview

Chapter 2, “Basic Event Generation” showed how to identify conditions and issue events to AutoOPERATOR from any MAINVIEW SRM function. The next step in the automation process is to respond to those events with a solution for the condition. AutoOPERATOR rules can be established to respond to the event, using data in event text to trigger automated solutions.

Enterprise Storage Automation provides solutions at the pool, volume, application, or data set level. Solutions are delivered through AUTO function FLST/RLST statements, which select the elements to be acted on and define the actions to be taken for the solution. The end result of a solution will be to either issue another event, perhaps to cause further analysis of the condition, or to submit a job to relieve the condition.

Solutions are requested through console commands to SVOS, which initiate the Enterprise Storage Automation AUTO functions and the associated FLST/RLSTs. This means that any vehicle that can issue console commands can request solutions of Enterprise Storage Automation.

Distributed sample solutions can be customized and implemented or you can create and implement your own solutions for use in any situation.

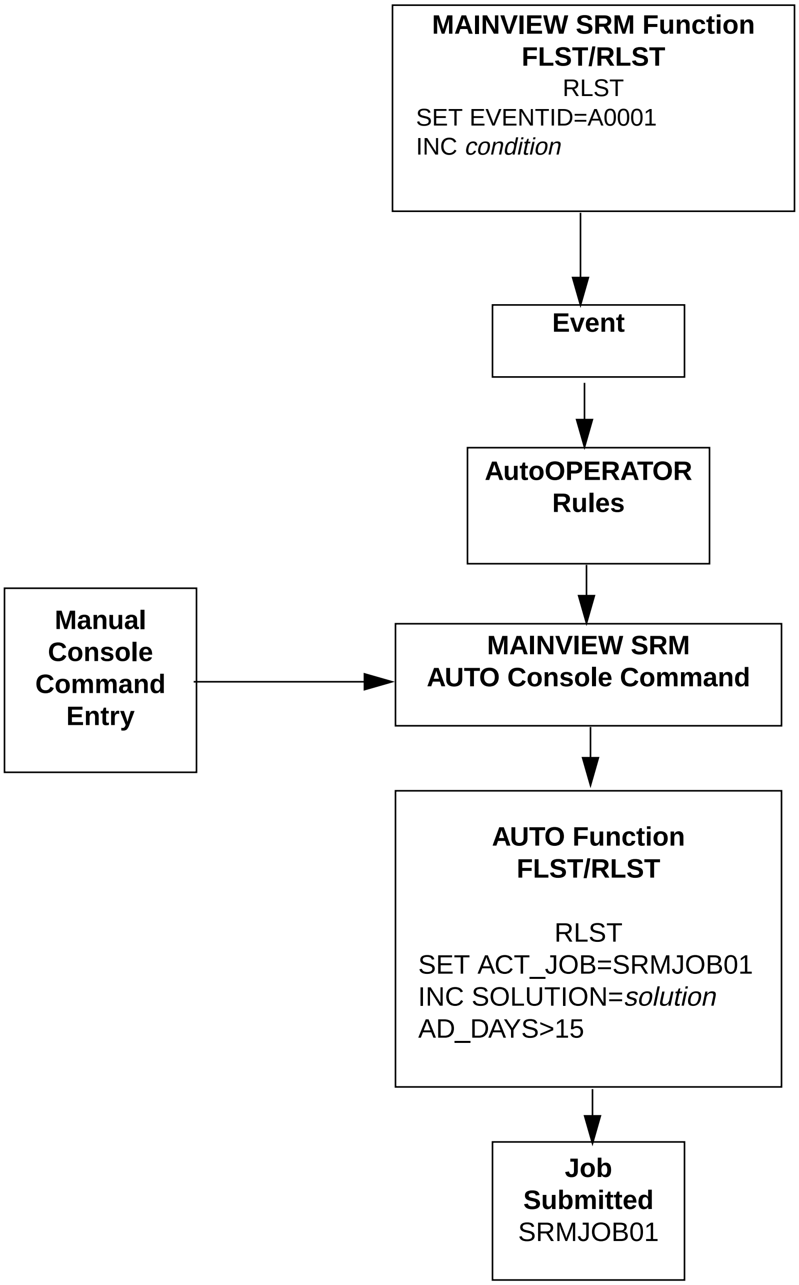
The AUTO functions can also be used to note storage conditions and issue events, performing the first step in the automation cycle, as well as providing the solution. This can be used as another early warning or problem detection capability. With the AUTO functions, pools, volumes, and applications can be monitored for almost any storage related condition, and an event can then be issued causing initiation of a solution for the condition.

The following topics are discussed in this chapter:

Automation Process Flow

Automation begins with basic event generation, where a situation is noted and a event sent to AutoOPERATOR. This step of noting the condition can be performed across all MAINVIEW SRM functions. AutoOPERATOR rules can be established that determine the action to take when the event is issued. This action can include issuing a OS/390 command, initiating an AUTO solution in response to the event as shown in Figure 3-1.

Figure 3-1 Automation Process Flow



Initiating Automation

Storage automation of pools, groups, volumes, and applications is initiated through a console command issued to the MAINVIEW SRM SVOS address space. You can issue the console command manually, through AutoOPERATOR rules, or through any other mechanism that allows the issuance of console commands (MAINVIEW Alarm, MAINVIEW API within AutoOPERATOR, and so on). Console commands are described in detail in Chapter 5, “Using AUTO Commands.”

AutoOPERATOR time-initiated rules can be used to initiate pool, volume, or application monitoring on a time interval basis. This is can be seen in the “Pool Space Management Automation Solution” on page 6-27

AUTO Functions and FLST/RLST Parameters

When a console command is issued in Enterprise Storage Automation, it initiates an AUTO function. Functions provide all the runtime services of MAINVIEW SRM. SMFUNC*xx* activates MAINVIEW SRM functions and controls message and tracing activity. SMFUNC*xx* points to members SMFLST*xx* and SMRLST*xx*, which select resources and control the operation of the functions.

Enterprise Storage Automation AUTO functions collect pool, volume, data set, and application space data and control the generation of events and the submission of jobs through FLST/RLST SET statements. This supplies a real-time collection interface with event generation and job submission capabilities for the automation of pool, volume, application, and data set resources.

For more information about using functions and SMFLST*xx* and SMRLST*xx* members, see the *MAINVIEW SRM User Guide and Reference*. For detailed information about Enterprise Storage Automation AUTO functions and FLST/RLST parameters, see Chapter 4, “Using AUTO Functions.”

Events

In Enterprise Storage Automation, the thresholds and conditions that generate user events are specified on INC/EXC statements in functions. User-defined storage occurrences can generate events in the form of messages. The event definition determines the text of the event, or message, sent to AutoOPERATOR. The EVENTID= parameter indicates the event that is to be fired when user-defined thresholds are exceeded.

All fields available in the FLST/RLST statements for the function are also available for insertion into the event text. In this way, you may customize the event and use the message text as a means to pass data from the function to AutoOPERATOR. The AutoOPERATOR rule can then use the data in the event text in triggering an automated solution in response to the event.

When a situation is noted in function FLST/RLST statements and an event is generated, the event ID may be unique to this condition or it may be generated in multiple FLST/RLST statements/conditions.

One determining factor in this decision is what solution do you want to take for the conditions. If the same solution is wanted for both conditions, a common event ID can be used. If separate solutions are needed for each condition, unique event IDs may be required. Also note, individual event IDs will be needed for each function if data is inserted into the event text. This is because each function has its own keywords that are allowed in the event text as substitutable variables.

AutoOPERATOR Rules

Responses to events are generated by AutoOPERATOR rules. AutoOPERATOR rules use data in event text to trigger automated solutions in response to events.

A rule is a two-part conditional statement. When the conditions specified in the first part of the statement are met, the actions in the second part of the statement are performed. Some rules use variables and some are time-initiated. Rules are organized into sets of related rules in the AutoOPERATOR Rules Processor. You can use AutoOPERATOR rules to further filter the event and initiate one or more solutions. A rule tells AutoOPERATOR to look for the event, usually by the first word of the event, which is always a message number for events sent from Enterprise Storage Automation (see “Defining the Event Using SMEVNTxx” on page 2-5). The rule can specify other filters on the event. For example, process an event/message only when the fourth word of the text contains POOL202 and the fifth word is greater than 80. The rule also specifies the action(s) to take when an event matching the criteria is seen. Actions include issuing OS/390 commands, setting variables, or invoking a REXX EXEC with parameters. There are other action possibilities; see the *MAINVIEW AutoOPERATOR Basic Automation Guide* for further information.

The command action of an AutoOPERATOR rule can be used to initiate a AUTO function solution in response to the problem or initiate an AUTO function that will further analyze the condition and generate other events depending on the outcome.

The MAINVIEW SRM distributed REXX EXEC can also be invoked to submit skeleton JCL in response to the event. Data from the event text can be set into variables for use in the skeleton JCL member. See “Skeleton Tailoring and Job Submission” on page 3-7

Suggested practice is to perform filtering in the FLST/RLST statements and have the rules just act on the event message number with no further filtering. This can reduce event activity, overhead, and can be easier to follow once implemented. The exception to this suggestion is in multi-level automation which is described in “Multi-Level Automation (MLA) Processing” on page 4-35. In that situation, the automation level is usually filtered in the AutoOPERATOR rule, where each rule initiates an action level.

Skeleton Tailoring and Job Submission

Many times the end result of a solution will be to submit a job. For example, one solution might cause the submission of a job to migrate selected data sets. This job JCL needs to specify the data set names selected by the solution in order to migrate them. This is made possible by the use of skeleton JCL.

AutoOPERATOR provides the skeleton JCL tailoring facility used by Enterprise Storage Automation. Any job submitted through Enterprise Storage Automation is done using skeleton JCL and the AutoOPERATOR Skeleton Tailoring Facility.

Skeleton tailoring processing involves the

- reading of a member of a partitioned data sets
- substitution of various variables
- processing of tailoring-specific directives
- ability to TSO submit the generated results

A variable is assumed whenever the variable recognition character is detected. This character defaults to ampersand sign (&) and can be changed for each member processed through the)DEFAULT directive. Characters that follow the & sign up to a blank or slash (\) are assumed to be the variable name.

Job submission can be accomplished with Enterprise Storage Automation in the following ways:

- AUTO function ACT_JOB keyword in the RLST member

  The ACT_JOB keyword is used to process a number of elements in a single job submission. For example, when selecting data sets to migrate, all data sets can be migrated in a single job. When ACT_JOB is used, Enterprise Storage Automation creates stem variables to contain the data from the function, such as data set names. AutoOPERATOR skeleton tailoring directives can be used in the skeleton JCL to replicate certain JCL statements for the number of elements (data sets) in the stem variable array. AutoOPERATOR skeleton tailoring variable substitution can be used within the repeated JCL statements to insert the data set name from the function into the JCL to be submitted.

  Note: ACT_JOB is required if performing multi-level automation. For more information, see “Multi-Level Automation (MLA) Processing” on page 4-35. If you use ACT_JOB, there are AutoOPERATOR rules that must be enabled to detect when the submitted job ends. These rules are distributed in Rule Set RULSRS01. For more information, see “Job Submission and Skeleton Tailoring” on page 4-29.

- REXX EXEC from an AutoOPERATOR rule

  You can invoke an Enterprise Storage Automation-supplied REXX EXEC from an AutoOPERATOR rule. This is intended to allow skeleton JCL and job submission capabilities from an event. This differs from the ACT_JOB RLST keyword in that a separate job is submitted for each event, so each job acts on a separate element. For example, using the data set migration solution, ACT_JOB would cause a single job, using skeleton JCL directive and variables to migrate the selected data sets. ACT_EVENTID could be used *in place of* ACT_JOB to issue an event for each data set. An AutoOPERATOR rule could then respond to the event by invoking EXEC CORSR005 (see “CORSR005 – Job Submission Using Console Commands” on page B-2). Data from the event text can be passed to the EXEC, which will create the same stem variables created by ACT_JOB. This method would cause a job to be submitted for each data set selected for migration, rather than one job for all selected data sets. The CORSR005 creates the same stem variables as the ACT_JOB keyword so that the same skeleton JCL can be used by ACT_JOB for a number of elements or by invoking CORSR005 for a single element.

For information about skeleton tailoring, see the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Advanced Multi-Level Automation

Multi-level automation refers to the process of exceeding a threshold, performing an action, checking the threshold again, performing another action, and so on until the threshold is no longer exceeded or until there are no other actions to take. For more information, see “Multi-Level Automation (MLA) Processing” on page 4-35.

Automation Views

Automation views allow you to see when events are being issued or jobs are being submitted, the connection status between Enterprise Storage Automation and the AutoOPERATOR subsystems, and statistics and status of the resources on which automation is or has been occurring.

Enterprise Storage Automation provides the following views:

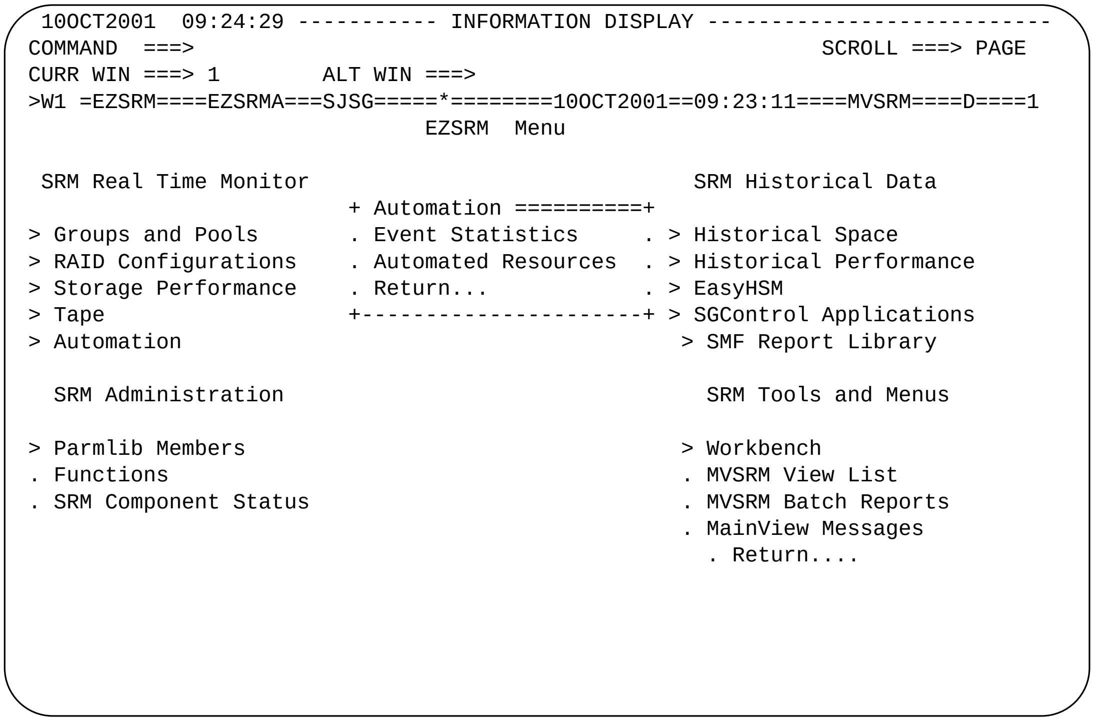
- Event Activity Statistics
- Automation Resources Statistics

To access Enterprise Storage Automation views:

Step 1 From the EZSRM Menu, select **Automation**.

The Automation pop-up menu is displayed in the center of the screen, as shown in Figure 3-2. View invocation and actions are described on the following pages.

Figure 3-2 EZSRM Menu, Automation Menu



```
 10OCT2001  09:24:29 ----------- INFORMATION DISPLAY --------------------------
COMMAND  ===>                                                 SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
>W1 =EZSRM====EZSRMA===SJSG=====*========10OCT2001==09:23:11====MVSRM====D====1
                               EZSRM  Menu

 SRM Real Time Monitor                              SRM Historical Data
                         + Automation ==========+
> Groups and Pools       . Event Statistics     . > Historical Space
> RAID Configurations    . Automated Resources  . > Historical Performance
> Storage Performance    . Return...            . > EasyHSM
> Tape                   +----------------------+ > SGControl Applications
> Automation                                       > SMF Report Library

  SRM Administration                                 SRM Tools and Menus

> Parmlib Members                                  > Workbench
. Functions                                        . MVSRM View List
. SRM Component Status                             . MVSRM Batch Reports
                                                   . MainView Messages
                                                     . Return....
```

Step 2 Select the menu option of your choice.

Step 3 To filter data, complete the fields provided on the data entry panel.

Step 4 Type **S** to the left of the **<==** symbol to process the request.

Table 3-1 describes the options that are available from the Automation pop-up menu.

Table 3-1 Enterprise Storage Automation Views

View	Description	View Name
Event Statistics	displays event statistics, the AutoOPERATOR subsystem, and the connection status to the AutoOPERATOR subsystem	EVENTS
Automation Resources Statistics	shows activity by the AUTOPOOL, AUTOVOL, AUTODS, and AUTOAPPL functions	AUTO

Enterprise Storage Automation (SVESA) must be active for either of these views to be available. Each view is shown and described in the pages that follow.

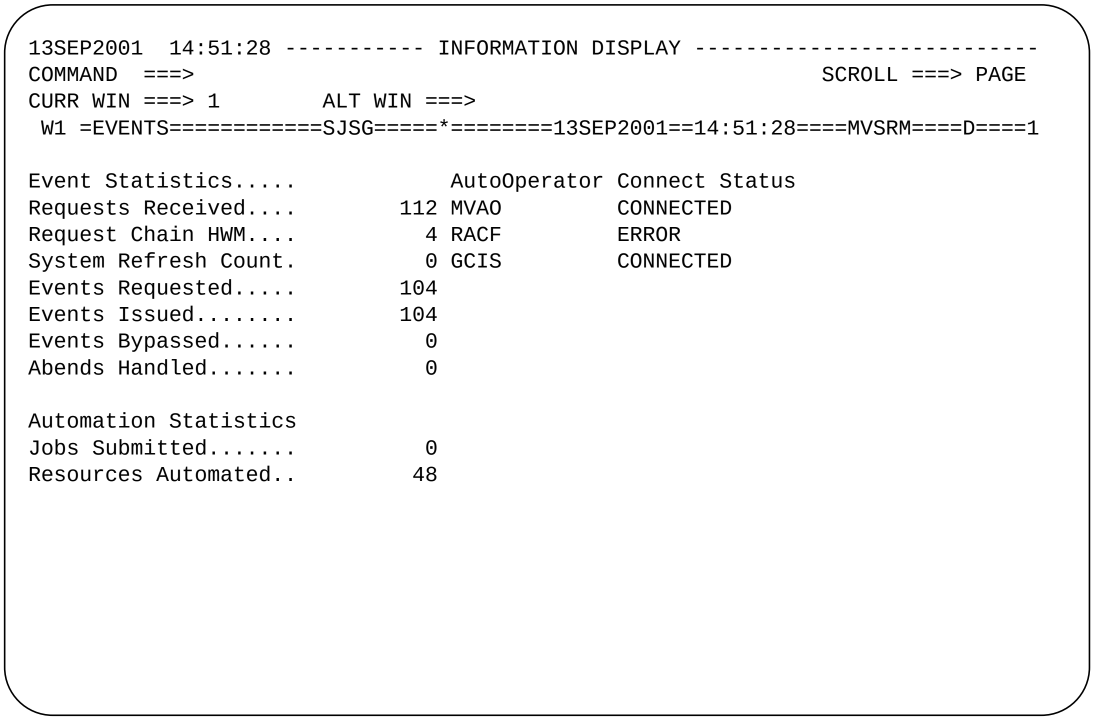
Event Statistics View

The Event Statistics view is a real-time view of event statistics and activity that you can use in conjunction with other metrics as a diagnostic or debugging aid. With this view, you can see when events are being issued. Some of these statistics are also displayed as joblog messages when Enterprise Storage Automation is stopped. Only overall counts are displayed; there is no log of events issued.

This view displays the

- Enterprise Storage Automation event statistics
- MAINVIEW SRM automation function statistics overview, which Hyperlinks to the detail information
- connection status between Enterprise Storage Automation and the AutoOPERATOR subsystems

Figure 3-3 Event Statistics View



```
13SEP2001  14:51:28 ----------- INFORMATION DISPLAY ---------------------------
COMMAND  ===>                                                 SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =EVENTS============SJSG=====*========13SEP2001==14:51:28====MVSRM====D====1

Event Statistics.....            AutoOperator Connect Status
Requests Received....        112 MVAO         CONNECTED
Request Chain HWM....          4 RACF         ERROR
System Refresh Count.          0 GCIS         CONNECTED
Events Requested.....        104
Events Issued........        104
Events Bypassed......          0
Abends Handled.......          0

Automation Statistics
Jobs Submitted.......          0
Resources Automated..         48
```

Table 3-2 describes the fields found on the Event Statistics view.

Table 3-2 Event Statistics View Field Descriptions

Field Name	Description
AutoOPERATOR	displays the name of the AutoOPERATOR subsystem defined in SMMSYS*xx* member to which Enterprise Storage Automation (SVESA) is to connect
Connect Status	displays the status of the AutoOPERATOR connection to Enterprise Storage Automation (SVESA)
Requests Received	displays the total number of work requests received by Enterprise Storage Automation during the current session These requests can include a stop request, event generation requests, and system requests.
Request Chain HWM	displays the number of request chains This indicates the high-water-mark of requests in the request chain for Enterprise Storage Automation during the current session. These requests can include a stop request, event generation requests, and system refresh requests. Each event generation request in the request chain represents approximately 1200 bytes of ECSA storage. There are several ways to reduce the number of requests in the request chain. You may want to remove some events from Enterprise storage Automation or add more filtering to event generation parameters so that requests are sent to the queue less often.
System Refresh Count	displays the total number of system refresh requests received by Enterprise Storage Automation during the current session
Events Requested	displays the total number of event generation requests received by Enterprise Storage Automation during the current session Event requests are generated by activated system events and by user-specified event generation parameters encountered during FLST/RLST function processing.
Events Issued	displays the total number of events sent to AutoOPERATOR during the current Enterprise Storage Automation session An event is sent to AutoOPERATOR when the DEST parameter on an active event definition specifies AOO as an event destination.
Events Bypassed	displays the number of events that were directed to AutoOPERATOR but were not sent because there were no active connections to an AutoOPERATOR subsystem
Abends Handled	displays the total number of abends encountered by Enterprise Storage Automation during the current session These abends were handled by Enterprise Storage Automation and may or may not cause Enterprise Storage Automation to stop. An abend that occurred during event request processing may account for any discrepancies between the number of event requests received and the number sent AutoOPERATOR.
Jobs Submitted	displays the number of batch jobs submitted by the automation processes
Resources Automated	displays the number of resources for which automation has processed requests

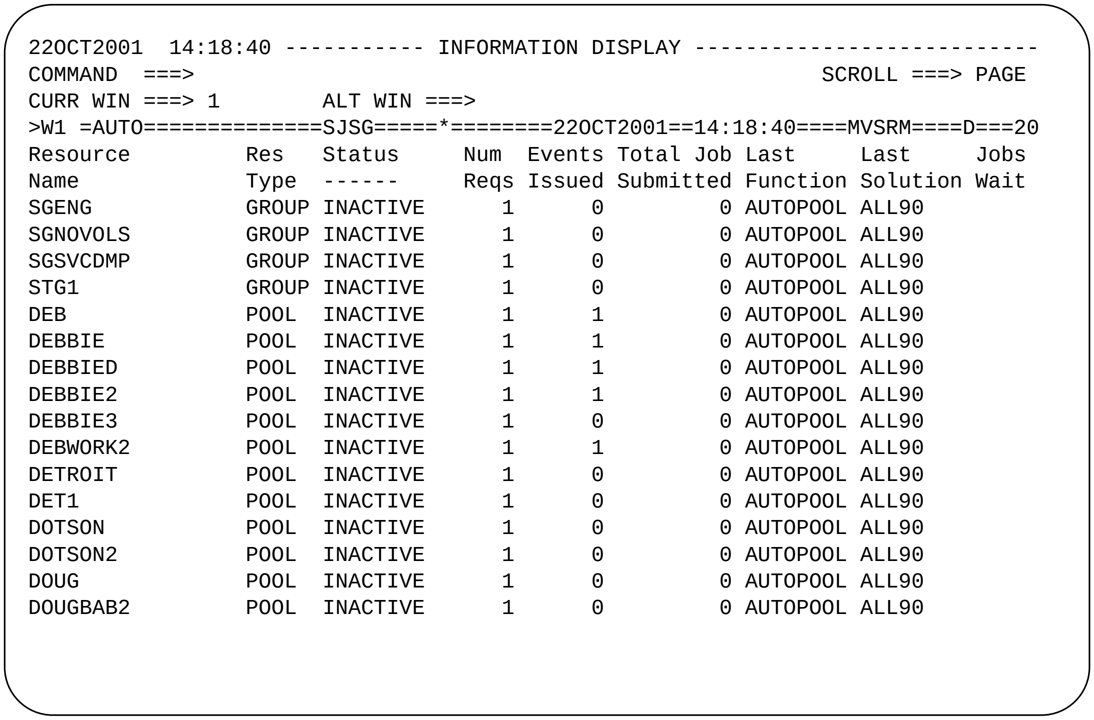
Automation Resource Statistics View

The Automation Resources Statistics view displays the statistics and status of the resources on which automation is or has been occurring. This view shows activity by the AUTOPOOL, AUTOVOL, AUTODS, and AUTOAPPL functions. A record on the view exists for each pool, volume, or application being monitored. Information on each resource includes:

- which resources are currently active
- the time automation started
- the current or last solution requested
- the level of the solution
- the current or last job submitted or event(s) issued
- the total number of jobs and events (actions) taken on the resource
- other statistics.

This view also displays the number of jobs outstanding for the resource.

Figure 3-4 Automation Resource Statistics View



```
22OCT2001  14:18:40 ----------- INFORMATION DISPLAY --------------------------
COMMAND  ===>                                                  SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
>W1 =AUTO==============SJSG=====*========22OCT2001==14:18:40====MVSRM====D===20
Resource         Res   Status     Num  Events Total Job Last     Last     Jobs
Name             Type  ------     Reqs Issued Submitted Function Solution Wait
SGENG            GROUP INACTIVE      1      0         0 AUTOPOOL ALL90
SGNOVOLS         GROUP INACTIVE      1      0         0 AUTOPOOL ALL90
SGSVCDMP         GROUP INACTIVE      1      0         0 AUTOPOOL ALL90
STG1             GROUP INACTIVE      1      0         0 AUTOPOOL ALL90
DEB              POOL  INACTIVE      1      1         0 AUTOPOOL ALL90
DEBBIE           POOL  INACTIVE      1      1         0 AUTOPOOL ALL90
DEBBIED          POOL  INACTIVE      1      1         0 AUTOPOOL ALL90
DEBBIE2          POOL  INACTIVE      1      1         0 AUTOPOOL ALL90
DEBBIE3          POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
DEBWORK2         POOL  INACTIVE      1      1         0 AUTOPOOL ALL90
DETROIT          POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
DET1             POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
DOTSON           POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
DOTSON2          POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
DOUG             POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
DOUGBAB2         POOL  INACTIVE      1      0         0 AUTOPOOL ALL90
```

Table 3-3 describes the fields found on the Automation Statistics view.

Table 3-3 Automation Resources Statistics View Field Definitions

Field Name	Description
Resource Name	displays the name of the resource being automated
Resource Type	displays the type of resource being automated Valid types: POOL, GROUP, SMS, VOL, APPL
Status	displays the status of the automation process for the named resource: • ACTIVE—resource is currently running an automation process • INACTIVE—resource has finished running an automation process • WAITING—resource is waiting for an automation process to complete
Num Reqs	the number of automation requests received for the named resource since the SVESA component was last started
MLA Status	States if the named resource is using multi-level automation.
Repeat Function	display the function that will be repeated for the cycle during multi-level automation: AUTOPOOL, AUTOVOL, AUTODS, AUTOAPPL
Repeat Resolution	displays the resolution that will be repeated for the cycle during multi-level automation Resolutions are customer defined.
Automation Level	displays the current level of automation for the resource A resource with a status of MLA=N will always display AUTOLEV0.
Number of Jobs Outstanding	displays the number of jobs requested and decrements the count as they complete if there are multiple jobs requested as part of the automation process
Total Events Issued	displays the total number of events issued for the named resource
Last Event ID Issued	displays the last event ID to be issued for the named resource
Last Event MSG ID Issued	displays the last event message ID that was issued for the named resource
Total Jobs Issued	displays the total number of jobs issued for the named resource
Last Job name Issued	displays the last job name issued during automation processing for the named resource
Date Requested	displays the date processing was initiated for the named resource
Time Requested	displays the time processing was initiated for the named resource
Date Ended	displays the date processing ended for the named resource
Time Ended	displays the time processing ended for the named resource
Last Solution	displays the name of the last solution executed for the named resource
Last Function	displays the name of the last function processed for the named resource
Max Autolev Value	displays the highest level of automation processing for the named resource

Chapter 4 Using AUTO Functions

The following topics provide detailed descriptions of functions that are used in Enterprise Storage Automation and the parameters that are used to define them.

Overview

MAINVIEW SRM storage management services are divided into functions. The SMFUNC*xx* parmlib member activates MAINVIEW SRM functions and controls message and tracing activity. Functions provide all the runtime services of MAINVIEW SRM. SMFUNC*xx* points to members SMFLST*xx* and SMRLST*xx*, which select resources and control the operation of the functions.

Enterprise Storage Automation provides the EVENTID=FLST/RLST statements to control event generation across all MAINVIEW SRM functions as described in Chapter 2, “Basic Event Generation.” Enterprise Storage Automation also provides AUTO functions FLST/RLST, which can be used to monitor pools, volumes, groups, and applications, and provide automated solution definitions to control the actions taken in response to storage conditions.

Unlike other MAINVIEW SRM functions, the AUTO functions are initiated through console commands, rather than during some other process or event. For example, the AUTOPOOL console command can be used to initiate pool automation. This console command can be issued by time-driven Rules in AutoOPERATOR or through any mechanism that can issue console commands.

For information on defining functions, system parameters, and FLST/RLST parameters see the *MAINVIEW SRM User Guide and Reference*.

SOLUTION Parameter

Like other MAINVIEW SRM functions, the AUTO function RLST members can, and probably will, contain multiple SET statements. In the AUTO functions, each SET statement first selects resources to process with INC/EXC statements and then determines the action(s) to take according to SET keywords in the statement. The solution parameter is used to associate an AUTO request with a particular SET statement, by specifying the solution value in the SOLUTION= keyword of the AUTO console command request and in INC statements for the SOLUTION parameter in the RLST member.

FLST/RLST conventions are such that a resource or record is checked against each SET statement in the RLST. When a resource matches a SET statement, the RLST process is stopped without interrogating subsequent SET statements. The resource is processed according to the matching SET statements keywords. With that in mind, consider multiple SET statements in the auto function FLST/RLST members.

Example

Consider these SET statements:

```
SET ACT_JOB=SRMRLSE
INC AD_DSORG = PO
    AD_PUSED < 80
SET ACT_JOB=SRMHMIG
INC AD_DAYS > 10
    AD_TRKSA > 100
```

The first SET statement is intended to release free space in PO type data sets; job SRMRLSE is submitted containing any data sets that have a DSORG of PO and have less than 80% space used in the data sets.

The second SET statement migrates data sets that have not been recently used, submitting job SRMHMIG to migrate any data sets that have not been referenced in at least 10 days and that are at least 100 tracks in size.

Note that the second SET statement will not process data sets that match the first SET statement. So, when trying to migrate data sets, any PO data sets that are less than 80% full will not be migrated.

The SOLUTION parameter is optional. Records for a request are not required to only match on one SET statement. Care must be given when coding RLST members when initiating multiple SET statements processing in a single request. Once a record matches a SET statement in the RLST, it is then associated with that SET statement. A record can only be associated with one SET statement. Any SET statements in the RLST member following the matching SET statement will never see this record. The result group created by the second SET statement could be incomplete if the record also applied to this SET statements INC/EXC statements.

SET Result Group

The AUTO functions RLST processes slightly differently than other MAINVIEW SRM functions. Most MAINVIEW SRM functions process a single resource or entity at a time, with the SET INC/EXC statements determining IF the resource is processed and the SET keywords determining HOW the resource is processed. With the AUTO functions RLST, each SET creates a group of records according to the INC/EXC statements. The AUTO function RLST processes a group of elements at a time, for example all data sets on a volume or in a pool. The INC/EXC statements determine which data sets in the pool will be acted upon. The SET statement keywords then determine how to process the GROUP of records selected. This group of records is called the *SET result group*.

The SET result group, those records matching the same SET statement, can be sorted using the SORT= keyword. This sorting is performed before the actions specified by ACT_JOB and/or ACT_EVENTID are taken, meaning that the ACT_JOB and/or ACT_EVENTID actions will be taken in sort order. When used in conjunction with ACT_COUNT, the action can be taken on the top *n* elements of the group; for example, the top 5 utilized volumes in a pool.

A request can create multiple SET result groups, although care should be taken when doing this. In the example above for SOLUTION, two SET result groups would be created, one with data sets to be compressed and one with data sets to migrate. Each SET result group would be processed individually according to its SET keywords.

AUTO Function Parameters

AUTO functions depend on the use of a variety of parameters. In Enterprise Storage Automation, a combination of system (global) parameters, Rules list parameters (RLST), and filter list parameters (FLST) are used to define the AUTOPOOL, AUTOVOL, AUTODS, and AUTOAPPL functions.

System Parameters

The parameters in SMMSYS*xx* that specifically affect automation functions are described in the following table:

Table 4-1 System Parameters

Parameter	Purpose
AUTO_MSGS=*Y/N*	controls whether the informational messages are produced to the SVOS joblog whenever an automation request for a resource is received
AUTO_MXTSK=*nn*	specifies the maximum number of tasks to be used in processing AUTO console command requests for any AUTO function If not specified, the default is 3. This value can be adjusted higher to decrease elapsed time of automation requests; however, this will increase the CPU and virtual storage utilization of the SVOS address space. If increased high enough, virtual storage in the SVOS address space will be completely utilized, resulting in space-related abends within SVOS processes. Care should be taken when adjusting this value.
VSCAN_MXTSK=*nn*	controls the number of tasks (TCBs) involved in scanning VTOCs for VTOC reporting and automation requests (2-30) If not specified, the default is 8. It can also be used to affect the response given to automation requests.

Rule List SET Parameters

All of the AUTO functions use the same Rule list parameters, which are described in this section. Filter list parameters are described after each function description.

SET Statement (in member SMRLST*xx*)

Parameter	Description
ACT_COUNT=*nnnnn*	specifies the maximum to the number of records to which actions can be taken
ACT_EVENTID=*event id*	specifies an event to issue for each record in the SET result group
ACT_JOB=*member name*	specifies the name of a member containing skeleton JCL to be submitted using the AutoOPERATOR Skeleton Tailoring Facility
ACT_SUM_FLD=*fldname*	causes a running total of the specified field to be maintained for each record against which a specified action is taken
ACT_SUM_LIM= *nnnnn*	used in conjunction with ACT_SUM_FLD to limit the number of records to be included in any specified action
SORT=(*fldname,x,fldname,x...*)	specifies the fields to sort prior to taking any actions on the group

ACT_COUNT=

Purpose: Optional parameter that specifies the maximum to the number of records to which actions can be taken. ACT_EVENTID and ACT_JOB keywords identify the possible actions to take. ACT_COUNT determines the maximum number of records in the SET result group for those actions. If ACT_COUNT is not specified, the action will be taken on all records in the SET result group, unless limited by use of ACT_SUM_LIM. Note that actions are taken after any sort requested with the SORT= keyword is performed.

Syntax: ACT_COUNT=*nnnnn*

where *nnnnn* is the maximum to the number of records to which actions can be taken (1–9999)

Default: None

ACT_EVENTID=

Purpose: Specifies an event to issue for each record in the SET result group. The action is applied after any SORT= specification is processed, which means the events will be generated in sort order. The number of events to issue can be limited by ACT_SUM_LIM or ACT_COUNT keywords.

ACT_EVENTID and ACT_JOB specify the action(s) to take. Every SET statement should have at least one of these keywords or no action will be taken. Both statements can be used; however, care should be taken when doing this.

EVENTID= cannot be used in AUTO Functions; ACT_EVENTID is used instead. The difference is that with ACT_EVENTID the event(s) are issued on a result group of records after any SORT= has been applied.

Syntax: ACT_EVENTID=*xxxxx*

where *xxxxx* is the 5-character value that corresponds to a valid event definition in parmlib member SMEVNT*xx*

Note: User events cannot begin with the letter I. These event IDs are reserved for MAINVIEW SRM internally-issued events. All distributed sample event definitions begin with the letter A.

An event is a message in which the ID corresponds to an SMEVNT*xx* entry, where customizable text can be specified.

Default: None

ACT_JOB=

Purpose: Specifies the name of a member containing skeleton JCL to be submitted using the AutoOPERATOR Skeleton Tailoring Facility. The skeleton tailoring facility is documented in the IMFEXEC TAILOR command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*. The member name must contain skeleton JCL and must be in UBBPROC data set, allocated in the SYSPROC DD statement in the AutoOPERATOR started task JCL.

Variables for each record in the result group are created for use in the skeleton JCL. The JCL member should make use of the)DO capability of skeleton tailoring to ensure that all stem values are processed by the submitted job. This is described in “Job Submission and Skeleton Tailoring” on page 4-29.

The action of creating variables for a record from the SET result group to be used in the job is done after any SORT= specification is processed on the SET result group. Just like ACT_EVENTID, this means the records will be included in sort order. The number of records to include in the job can be limited by ACT_SUM_LIM or ACT_COUNT keywords.

ACT_JOB is recommended over ACT_EVENTID if the result of the ACT_EVENTID will be to submit a job. Using ACT_EVENTID in conjunction with AutoOPERATOR Rules to submit a job causes a job to be submitted for each record in the FLST/RLST SET result group (as limited by a ACT_COUNT or ACT_SUM_LIM). Using ACT_JOB causes all of the SET result records to be passed to the skeleton tailoring facility in a single iteration using stem variables.

When ACT_JOB is used, a set of AutoOPERATOR Rules that detect job end must be enabled. These Rules are in distributed Rule Set RULSRS01. The jobname of the submitted job must be filtered on in the Rules within this Rule Set. See “Job Submission and Skeleton Tailoring” on page 4-29.

ACT_JOB and ACT_EVENTID specify the action(s) to take. Every SET statement should have at least one of these keywords, or no action will be taken. Both statements can be used, however, care should be taken when doing this.

Syntax: ACT_JOB=*xxxxxxxx*

where *xxxxxxxx* is the 1–8 character name of the member that contains the skeleton JCL in the UBBPROC data set in the AO started task

Default: None

ACT_SUM_FLD=

Purpose: Optional parameter that causes a running total of the specified field to be maintained for each record against which a specified action is taken. When using ACT_SUM_FLD, ACT_SUM_LIM can be used to stop action from being taken once the running total reaches a specified value.

Syntax: ACT_SUM_FLD=*xxxxxxxxxxx*

where *xxxxxxxxxxx* is a 1–11-character valid INC/EXC field name for the function associated with the RLST; the field name must be that of a numeric field

Default: None

ACT_SUM_LIM=

Purpose: Optional parameter that can be used to limit the number of records to be included in any specified action. If specified, ACT_SUM_FLD= should also be specified. A running total of the field specified in ACT_SUM_FLD is maintained as each action is taken. At the completion of the action the total is updated. Prior to taking the next action to total is compared to ACT_SUM_LIM. If the total is less than the limit, the next action is taken. If the total is equal to or greater than the limit, action processing is bypassed for the remainder of the records in the SET result group. ACT_SUM_LIM is not a count of records being acted upon. To limit by a count of records see ACT_COUNT.

Syntax: ACT_SUM_LIM=*nnnnnnnnnn*

where *nnnnnnnnnn* is the maximum value reached by ACT_SUM_FLD (1–9223372036854775807)

Default: None

SORT=

Purpose: Specifies INC/EXC fields to sort the result group of records. The fields specified must be valid INC/EXC fields for the function. Any ACT_JOB or ACT_EVENTID actions are taken after this sort is applied, and therefore will be taken in sort order.

This can be used, in conjunction with ACT_COUNT to take actions on the resources with the highest or lowest values in any of the information.

For example: SORT=(AD_DAYS,D) in the AUTODS function sorts a SET result group of data set records by the number of days since the data set was last referenced. This, along with ACT_COUNT=5, causes any specified action to be taken on the 5 data sets that have been referenced the least recently.

For more information, see the Usage Notes for the specific function.

Syntax: SORT=(*fieldname*,*x*,*fldname*,*x*,*….*)

where *fieldname* is a valid INC/EXC field for the function, and *x* is either A (ascending) or D (descending); can contain a maximum of 4 field and direction pairs; the total length of the specified sort fields cannot exceed 100 bytes

Default: None. If not specified, the result group of records is not sorted and actions are taken in the order the information was collected.

AUTOPOOL Function – Pool Automation

The AUTOPOOL function is used to initiate pool automation for pools, SMS Storage Groups, or SMS Pools (see AUTOVOL and AUTOAPPL for automation of volume lists or SG-Control applications). Pool automation provides the capability to interrogate space or other pool attributes and initiate actions when one or any number of thresholds are exceeded. Actions include:

- submitting a job using a skeleton tailoring facility, similar to ISPF tailoring, to perform a batch process on the pool
- issuing events to communicate with AutoOPERATOR, which allows all the actions available in AutoOPERATOR to be initiated in response to a pool condition.

In conjunction with events being issued, AutoOPERATOR Rules can be used to initiate volume automation solutions for volumes in the pool or data set automation solutions for data sets in the pool. These solutions can be initiated in a tiered approach where automated solutions can be applied to a pool in steps, which are defined in multi-level processing. The multi-level process generates an event caused by an exceeded threshold, then takes an action, then checks the threshold again, then takes another action, and so on. For further information, see “Multi-Level Automation (MLA) Processing” on page 4-35

The AUTOPOOL FLST/RLST statements are used to detect situations or thresholds and specify the event or job actions to take. Space, status, and other pool attributes fields can be interrogated to determine when actions should be initiated.

Unlike other MAINVIEW SRM functions, this function is initiated through user request rather than during some other process or event. The AUTOPOOL console command can be used to initiate volume automation on a list of individual volumes or on volumes in a pool (see the AUTOPOOL command). This console command can be issued by time-driven Rules in AutoOPERATOR or through any mechanism that can issue console commands.

Table 4-2 Filter List Parameters for AUTOPOOL (Part 1 of 3)

Parameter	Description
AP_CTIGC=*nnnnnnnnnnn*	specifies the largest contiguous free cylinders (0-2147483647)
AP_CTIGT=*nnnnnnnnnnn*	specifies the largest contiguous free tracks (0-2147483647)
AP_FREEC=*nnnnnnnnnnn*	specifies the total count of free cylinders (0-2147483647)

Table 4-2 Filter List Parameters for AUTOPOOL (Part 2 of 3)

Parameter	Description
AP_FREED=*nnnnnnnnnnn*	specifies the total number of free data set control blocks (DSCB) for all volumes (0-2147483647)
AP_FREET=*nnnnnnnnnnn*	specifies the total count of free tracks (0-2147483647)
AP_FREEV=*nnnnnnnnnnn*	specifies the total number of free index records in a volume’s VTOC index (0-2147483647)
AP_FREEX=*nnnnnnnnnn*	specifies the total number of free extents (0-2147483647)
AP_FSIZE=*nnnnnnnnnn*	specifies the free size (in MB) of unused space in the pool (0-2147483647)
AP_HFULL=*nnn*	specifies the high-water mark volume percentage full, which is the percentage full of the highest utilized volume in the pool (0-100)
AP_HREEC=*nnnnnnnnnn*	specifies the high-water mark count of free cylinders (0-2147483647)
AP_HREED=*nnnnnnnnnn*	specifies the high-water mark count of free DSCBs (0-2147483647)
AP_HREET=*nnnnnnnnnn*	specifies the high-water mark count of free tracks (0-2147483647)
AP_HREEV=*nnnnnnnnnn*	specifies the high-water mark count of free VIR (0-2147483647)
AP_HREEX=*nnnnnnnnnn*	specifies the high-water mark count of free extent (0-2147483647)
AP_HVFRG=*nnn*	specifies the high-water mark fragmentation index (0-100)
AP_HVFUL=*nnn*	specifies the high-water mark VTOC percentage full (0-100)
AP_LFULL=*nnn*	specifies the low-water mark volume percentage full (0-100)
AP_LPRIC=*nnnnnnnnnn*	specifies the largest primary allocation (cylinders) (0-2147483647)
AP_LPRIT=*nnnnnnnnnn*	specifies the largest primary allocation (tracks) (0-2147483647)
AP_LREEC=*nnnnnnnnnn*	specifies the low-water mark count of free cylinders (0-2147483647)
AP_LREED=*nnnnnnnnnn*	specifies the low-water mark count of free DSCBs (0-2147483647)
AP_LREET=*nnnnnnnnnn*	specifies the low-water mark count of free tracks (0-2147483647)
AP_LREEV=*nnnnnnnnnn*	specifies the low-water mark count of free VIR (0-2147483647)
AP_LREEX=*nnnnnnnnnn*	specifies the low-water mark count of free extent (0-2147483647)
AP_LVFRG=*nnnnnnnnnn*	specifies the low-water mark fragmentation index (0-100)
AP_LVFUL=*nnn*	specifies the low-water mark of VTOC percentage full (0-100)
AP_PERFL=*nnn*	specifies the pool percentage full based on used space as it relates to total space (0-100)
AP_POOL=*xxxxxxxxx*	specifies the pool, group, or SMS pool name (1-30)
AP_TSIZE=*nnnnnnnnnn*	specifies the total size (in MB) of space in the pool (0-2147483647)
AP_TYPE=*xxxxxxx*	specifies the pool type Values are: POOL, SMSPOOL, or GROUP.
AP_USIZE=*nnnnnnnnnn*	specifies the amount of allocated space in the pool (used size) (0-2147483647)
AP_VOLC=*nnnnnnnnnn*	specifies the number of online volumes in this pool on the collecting OS/390 system (0-2147483647)

Table 4-2 Filter List Parameters for AUTOPOOL (Part 3 of 3)

Parameter	Description
AP_VOLD=*nnnnnnnnnn*	specifies the volume drop count (due to errors) (0-2147483647) Number of volumes in the pool that were not collected due to collection errors. The totals for the pool may be invalid due to these volumes not be included. The SVOS joblog will contain error messages indicating the errors encountered.
AUTOLEV=*xxxxxxxx*	contains an 8-character literal AUTOLEV*x*, where *x* is a number indicating the current automation level for the resource being automated AUTOLEV0 indicates that multiple levels are not being used. AUTOLEV1 indicates the first level of a multiple level request. For multiple level automation this field must be used either in this FLST/RLST member or contained within event text of any event issued and then referenced in the AutoOPERATOR Rule that looks for the event.
SOLUTION=*xxxxxxxx*	contains the solution value from the originating AUTOVOL command (1-8 characters) It is recommended that each SET statement in the AUTOVOL function include a unique solution value. This solution value can then be used on the AUTOVOL console command to invoke the solution. For more information, see “SOLUTION Parameter” on page 4-2.

AUTOPOOL Variables for Skeleton Tailoring

Variables created by the AUTOPOOL function for use in skeleton JCL are shown in Table 4-3. These variables can be used in skeleton JCL when it is submitted using the ACT_JOB keyword in the AUTOPOOL function. Other functions create a slightly different set of variables.

Table 4-3 AUTOPOOL Variables for Skeleton Tailoring

Stem Variable	Description
&POOL.0 or &GROUP.0 or &SMSPOOL.0	one of these stem variables will be created, dependent on the resource being processed by the AUTOPOOL function The variable contains the number of entry names in the pool, group, or SMS pool array.
&POOL.*n* or &GROUP.*n* or &SMSPOOL.*n*	where *n* is a number from 1 to the value in &POOL.0 or &GROUP.0 or &SMSPOOL.0 The variable contains a pool name generated by the AUTOPOOL function that should be included in the skeleton JCL using the)DO directive statement.
&RESTYPE.*n*	number of resource types in variable The variable contains the resource type or *pool*, *group*, or *SMS pool*. This can be used as documentation in the skeleton JCL.

Table 4-3 AUTOPOOL Variables for Skeleton Tailoring

Stem Variable	Description
&FUNC.*n*	contains number, will be same a &POOL.*n* &FUNC.*n* contains the AUTO function causing the job submission, in this case AUTOPOOL. Created as stem, all entries will be the same value. This can be used as documentation in the skeleton JCL.
&SOL.*n*	contains the same number as &POOL.*n* &SOL.*n* contains a solution causing the job submission. All entries will be the same value. This can be used as documentation in the skeleton JCL. For more information, see “SOLUTION Parameter” on page 4-2.

AUTOPOOL Function Usage Notes

The SET result group contains pool space information for each pool in the automation request. Sorting this information causes the pools in the request to be sorted. For example, the following request is made:

```
SVOS AUTOPOOL Pool=(PoolA,PoolB,PoolC)
```

Space information for PoolA, PoolB, and PoolC is obtained and run through FLST. The SET result group would then contain entries for PoolA, PoolB, and PoolC, assuming the all matched a SET statement.

```
SET SORT=(AP_PERFL,D)                             x
    ACT_COUNT=2                                   x
    ACT_EVENTID=AP009
INC AP_POOL=POOL*                                 x
    SOLUTION=AP009ALL
```

In this example, the SORT= causes the pool space entries to be sorted by percent full, resulting in the pools with the largest percent full at the top. Event AP009 would then be issued for the two pools, due to ACT_COUNT=2. In essence this has caused an event to be issued for the two highest utilized pools in the request.

The pools included in a single request determine the effects of SORT and ACT_COUNT and ACT_SUM_LIM keywords. This sort statement is ignored for AUTOPOOL command requests that have only one pool match the INC/EXC statements for the SET or that specify a single pool name, such as `SVOS AUTOPOOL POOL=MYPOOL,SOLUTION=AP009ALL`.

AUTOVOL Function – Volume Automation

The AUTOVOL function is used to initiate automation for a volume or pool of volumes. Volume automation provides the capability to interrogate space or other volume attributes and initiate actions when one or any number of thresholds are exceeded. Actions include:

- submitting a job using a skeleton tailoring facility, similar to ISPF tailoring, to perform a batch process on the volume or volumes
- issuing events to communicate with AutoOPERATOR, allows all the actions available in AutoOPERATOR to be initiated in response to a volume condition.

In conjunction with events being issued, AutoOPERATOR Rules can be used to initiate data set-level solutions for the selected volumes. These solutions can be taken step by step, checking the volume utilization at the end of each step.

The AUTOVOL FLST/RLST statements are used to detect situations or thresholds and specify the event or job actions to take. A number of volume space information fields can be interrogated to determine when actions should be initiated.

Unlike other MAINVIEW SRM functions, this function is initiated through user request rather than during some other process or event. The AUTOVOL console command can be used to initiate volume automation on a list of individual volumes or on volumes in a pool (see the AUTOVOL command). This console command can be issued by time-driven Rules in AutoOPERATOR or through any mechanism that can issue console commands.

Table 4-4 Filter List Parameters for AUTOVOL (Part 1 of 3)

Parameter	Definition
AUTOLEV=*xxxxxxxx*	contains an 8-character literal AUTOLEV*x*, where *x* is a number indicating the current automation level for the resource being automated AUTOLEV0 indicates that multiple levels are not being used. AUTOLEV1 indicates the first level of a multiple level request. For multiple level automation this field must be used either in this FLST/RLST member or contained within event text of any event issued and then referenced in the AutoOPERATOR Rule that looks for the event.
AV_CTIGC=*nnnnnnnnnn*	contains the largest single extent in full cylinders available for allocation (0-2147483647)
AV_CTIGT=*nnnnnnnnnn*	contains the largest single extent in tracks available for allocation (0-2147483647)
AV_DEV=*xxxxxxxx*	contains the unit control block address for the volume (1-8 characters)

Table 4-4 Filter List Parameters for AUTOVOL (Part 2 of 3)

Parameter	Definition
AV_FRAGI=*nnnnnnnnnn*	contains the fragmentation index value of the volume (0-2147483647) The higher the value, the more fragmented the volume.
AV_FREEC=*nnnnnnnnnn*	contains the number of free cylinders on the volume (0-2147483647)
AV_FREED=*nnnnnnnnnn*	contains the number of free (Format 0) DSCBs on the volume (0-2147483647)
AV_FREET=*nnnnnnnnnn*	contains the number of free tracks on the volume (0-2147483647)
AV_FREEV=*nnnnnnnnnn*	contains the number of free VIRs (VTOC index records) on the volume (0-2147483647)
AV_FREEX=*nnnnnnnnnn*	total amount of free extents on the volume (0-2147483647)
AV_FSIZE=*nnnnnnnnnn*	amount of space not used on the volume in megabytes (0-2147483647)
AV_FULL=*nnn*	contains the percentage of used space to total space for the volume (0-100) This is a whole number, no decimal places.
AV_LPRIC=*nnnnnnnnnn*	contains the largest possible primary extent in cylinders (0-2147483647) This is the sum of the 5 largest extents on the volume.
AV_LPRIT=*nnnnnnnnnn*	contains the largest possible primary extent in tracks (0-2147483647) This is the sum of the 5 largest extents on the volume.
AV_MNT=*xxx*	contains how the volume is mounted Indicators are: PUB Public PVT Private STG Storage SYS System
AV_POOL=*xxxxxxxx*	pool name displayed if the AUTOVOL function is associated with an AUTOPOOL POOL= function (1-8 characters)
AV_SMSGP=*xxxxxxxx*	contains the SMS -assigned storage group name (1-8 characters) If the volume is not SMS managed, this field will be blank.

AV_SMSI=*xx*	contains the SMS status of the volume
	Values are:
	M = SMS managed
	QA = SMS quiesced all
	QN = SMS quiesced new
	DA = SMS disabled all
	DN = SMS disabled new
	UN = Not SMS managed
	NA = Unknown

AV_SPOOL=*xxxxxxxx*	SMS pool name displayed if the AUTOVOL function is associated with an AUTOPOOL SMSPOOL= function (1-8 characters)
AV_TSIZE=*nnnnnnnnnn*	total volume size in megabytes (0-2147483647)
AV_USIZE=*nnnnnnnnnn*	amount of space used on the volume in megabytes (0-2147483647)
AV_VOL=*xxxxxx*	contains the volume serial number (1-6 characters)
AV_VTOCF=*nnn*	contains the percentage of used VTOC space (0-100) This percentage is a whole number, no decimal places.

Table 4-4 Filter List Parameters for AUTOVOL (Part 3 of 3)

Parameter	Definition
AV_VTOCI=*xxx*	contains the VTOC index status Values are: ACT = VTOC Index is defined and active INA = VTOC Index is defined but is not active UND = VTOC Index is not defined
AV_VTOCZ=*nnnnnnnnnn*	contains the volumes VTOC size in tracks (0-2147483647)
SOLUTION=*xxxxxxxx*	contains the solution value from the originating AUTOVOL command (1-8 characters) It is recommended that each SET statement in the AUTOVOL function include a unique solution value. This solution value can then be used on the AUTOVOL console command to invoke the solution. For more information, see “SOLUTION Parameter” on page 4-2.

AUTOVOL Variables for Skeleton Tailoring

Variables created by the AUTOVOL function for use in skeleton JCL are shown in Table 4-5. These variables can be used in skeleton JCL when it is submitted using the ACT_JOB keyword in the AUTOVOL function. Other functions create a slightly different set of variables.

Table 4-5 AUTOVOL Variables for Skeleton Tailoring

Resources	Variable	Description
Volume Resource (AUTOVOL VOL=(*list*)	&VOLSER.*n*	contains the volume serials generated by the AUTOVOL function that should be included in the skeleton JCL
	&DEVT.*n*	contains the device type of the volume This is the 8 byte device type, for example 3390-A34.
	&RESTYPE.*n*	indicates the resource type All array entries with the same value will contain one of the following values: pool, group, SMS pool–indicate AUTOVOL pool=/group=/and so on appl–indicates AUTOVOL APPL= was used volume–indicates AUTOVOL VOL= was used
	&FUNC.*n*	contains number, will be same a &VOL.*n* &FUNC.*n* contains the AUTO function causing the job submission, in this case AUTOPOOL. Created as stem, all entries will be the same value. This can be used as documentation in the skeleton JCL.
	&SOL.*n*	contains the same number as &VOL.*n* &SOL.*n* contains a solution causing the job submission. All entries will be the same value. This can be used as documentation in the skeleton JCL.

Table 4-5 AUTOVOL Variables for Skeleton Tailoring

Resources	Variable	Description
Pool Resources (AUTOVOL POOL/GROUP/SMSPOOL=*poolname*)	&VOLSER.*n*	contains the volume serials generated by the AUTOVOL function that should be included in the skeleton JCL
	&DEVT.*n*	contains the device type of the volume This is the 8 byte device type, for example 3390-A34.
	&RESTYPE.*n*	indicates the resource type All array entries with the same value will contain one of the following values: pool, group, SMS pool–indicate AUTOVOL pool=/group=/and so on appl–indicates AUTOVOL APPL= was used volume–indicates AUTOVOL VOL= was used
	&FUNC.*n*	contains number, will be same a &VOL.*n* &FUNC.*n* contains the AUTO function causing the job submission, in this case AUTOPOOL. Created as stem, all entries will be the same value. This can be used as documentation in the skeleton JCL.
	&SOL.*n*	contains the same number as &VOL.*n* &SOL.*n* contains a solution causing the job submission. All entries will be the same value. This can be used as documentation in the skeleton JCL.
	&POOL.*n*, &GROUP.*n*, or &SMSPOOL.*n*	resource, depending on which was used

AUTOVOL Function Usage Notes

The SET result group will contain space information for either all the volumes in a pool, if AUTOVOL Pool= was used, or all the volumes in the request, if AUTOVOL Vol=(volume list) was used. For example, the following request is made:

```
SVOS AUTOVOL Pool=PoolA
```

Space information for all the volumes in PoolA is obtained and run through FLST. The SET result group will include any volumes in the pool that matched a RLST SET statement. Sorting this list, for example by percent full, and issuing events or submitting jobs would cause actions on the specified number of most utilized volumes in the pool.

If the following command is used

```
SVOS AUTOVOL Vol=(VolA,VolB,VolC)
```

space information for the volumes requested, is obtained and run through FLST. The SET result group will include the volumes from this list that matched a RLST SET statement. Similar to AUTOPOOL and AUTOAPPL, the volumes in the request determine the result group. For Vol= requests that specify a single volume, the SORT and ACT_SUM_LIM RLST keywords have no effect.

AUTODS Function – Data Set Automation

The AUTODS function is used to initiate automation for a data set. The AUTODS process collects data set information for all the data sets on the entity passed and runs them through the AUTODS FLST/RLST. The AUTODS function is not invoked unless requested. Actions include:

- submitting a job using a skeleton tailoring facility, similar to ISPF tailoring, to perform a batch process on the data set
- issuing events using the Enterprise Storage Automation component to communicate with AutoOPERATOR, allows all the actions available in AutoOPERATOR to be initiated in response to a data set condition.

AUTODS is used to deliver data set-level automated solutions in response to any storage condition. Initiated through console commands, these solutions can be taken step by step, checking the data set condition at the end of each step.

The AUTODS FLST/RLST statements are used to select the data sets to be acted on in the solution. Almost any data set attribute may be interrogated to either include or exclude the data set from the solution.

Unlike other MAINVIEW SRM functions, this function is initiated through user request rather than during some other process or event. The AUTODS console command can be used to initiate data set automation. This console command can be issued by time-driven Rules within AutoOPERATOR or through any mechanism that can issue console commands.

Table 4-6 Filter List Parameters for AUTODS (Part 1 of 4)

Parameter	Definition
AD_ALVL1=*xxxxxxxxxxxxxxxx*	contains the first 16 characters of the application name (SG-Control Application Level 1)
AD_ALVL2=*xxxxxxxxxxxxxxxx*	contains the first 16 characters of the application name (SG-Control Application Level 2)
AD_ALVL3=*xxxxxxxxxxxxxxxx*	contains the first 16 characters of the application name (SG-Control Application Level 3)
AD_ALVL4=*xxxxxxxxxxxxxxxx*	contains the first 16 characters of the application name (SG-Control Application Level 4)
AD_BLKEF=*nnn*	specifies the percentage of blocking efficiency (0-100)
AD_BLKSZ=*nnnnn*	contains physical block size or VSAM control interval size for the data set (0-32760) For VSAM KSDS, ESDS and RRDS, the control interval size is displayed. For all other data set types, the physical block size is displayed. For VSAM data sets, the physical block size and control interval size are obtained from the catalog.

Table 4-6 Filter List Parameters for AUTODS (Part 2 of 4)

Parameter	Definition
AD_BLKTR=*nnnnnnnnnn*	contains the number of physical blocks that will fit on one track (0-2147483647) The value relates to the Percent Efficiency value
AD_CASPL=*nnnnnnnnnn*	contains the number of VSAM control area splits performed on the data set (0-2147483647) This number is for the entire VSAM data set if it spans multiple volumes. Non-VSAM data sets will show NA in this field.
AD_CAT=*x*	contains the data set's catalog status. Indicators are: C = The data set is cataloged and resides on the volume shown. N = The data set is not cataloged, but resides on the volume shown. D = The data set is not cataloged, but resides on the volume shown. However, there is a data set with the same name on a different volume that is cataloged. U = It is unknown whether the data set is cataloged, but it does reside on the volume shown.
AD_CDATE=*xxxxxxxxxx*	contains the 10-character creation date of the data set in yyyy/mm/dd format
AD_CHG=*Y/N*	contains an indicator of whether the data set has been opened for output (changed)
AD_CISPL=*nnnnnnnnnn*	contains the number of VSAM control interval splits performed on the data set (0-2147483647) This number is for the entire VSAM data set if it spans multiple volumes. Non-VSAM data sets will show NA in this field.
AD_DAYS=*nnnnnnnnnn*	contains the number of days since the data set was opened (0-2147483647) This number is calculated by subtracting the last reference date from the current date. If the data set was never opened, this field will contain zero.
AD_DCLAS=*xxxxxxxx*	contains the SMS-assigned data class or one of the following values for the data set: NONE = The data set is cataloged and non-SMS managed. DUPLIC = The data set is a duplicate (not cataloged). UNDET = The data set's SMS status could not be determined
AD_DSN=*xxxxxxxxxx*	specifies the data set name (1-44)

AD_DSORG=*xxxx*	contains the data set file organization and access method used to manage the data set. The indicators and their meanings are:
	PS = Physical Sequential (QSAM)
	PO = Partitioned Data Set (BPAM)
	PDSE = Partitioned Data Set Extended (LIBRARY)
	VS = VSAM
	DA = Direct Access (BDAM)
	IS = Indexed Sequential (ISAM)
	– = The data set organization could not be determined or the data set was never opened.

AD_EXTS=*nnnnnnnnnn*	contains the number of extents occupied by the data set on the volume (0-2147483647)
AD_GROUP=*xxxxxxxxxx*	SMS storage group name displayed if the AUTODS function is associated with an AUTOPOOL GROUP= function (1-30 characters)

Table 4-6 Filter List Parameters for AUTODS (Part 3 of 4)

Parameter	Definition
AD_LDATE=*xxxxxxxxxx*	contains the last date the data set was opened in yyyy/mm/dd format If the data set was never opened, this field will contain blanks.
AD_LRECL=*nnnnn*	contains the maximum record length for the data set (0-99999) For VSAM data sets, the record length is obtained from the catalog.
AD_MCLAS=*xxxxxxxx*	contains the SMS-assigned management class or one of the following values: NONE = The data set is cataloged and non-SMS managed. DUPLIC = The data set is a duplicate (not cataloged). UNDET = The data set's SMS status could not be determined
AD_POOL=*xxxxxxxx*	pool name displayed if the AUTODS function is associated with an AUTOPOOL POOL= function (1-8 characters)
AD_PUSED=*nnn*	contains the percentage of allocation that is used (0-100)
AD_REBLK=*Y/N*	contains the reblockable indicator, which determines whether the data set can be reblocked by the system when being moved from one device geometry to another This is also known as System Determined Blocksize and typically allocates a blocksize that uses the space on the device most efficiently. N indicates that the data set is not allocated with System Determined Blocksizing and must be manually reblocked when moved from one device geometry to another.
AD_RECFM=*xxxxx*	contains the data set record format The record format indicates the type of record access along with the general format of the records and blocks. Indicator meanings are: **Non-VSAM data sets:** F = Fixed length blocks V = Variable length blocks U = Undefined block lengths B = Records are blocked S = Records span multiple blocks M = Records contain machine control characters A = Records contain ANSI printer control characters ---- = The data set organization could not be determined or the data set was never opened. **VSAM data sets:** ESDS = Entry-Sequenced data set KSDS = Key-Sequenced data set LDS = Linear data set PAGE = System page data set UCAT = User catalog VVDS = ICF catalog system data set ---- = The data set organization could not be determined or the data set was never opened.
AD_SCLAS=*xxxxxx*	contains the SMS-assigned storage class or one of the following values: NONE = The data set is cataloged and non-SMS managed. DUPLIC = The data set is a duplicate (not cataloged). UNDET = The data set's SMS status could not be determined

Table 4-6 Filter List Parameters for AUTODS (Part 4 of 4)

Parameter	Definition
AD_SIZE=*nnnnnnnnnn*	contains the data set size in kilobytes (one kilobyte equals 1024 bytes) on the volume (0-2147483647)

AD_SMSI=*xx*	specifies the SMS status of the volume
	Values are:
	M = SMS managed
	QA = SMS quiesced all
	QN = SMS quiesced new
	DA = SMS disabled all
	DN = SMS disabled new
	UN = Not SMS managed
	NA = Unknown

AD_SPOOL=*xxxxxxxx*	specifies the SMS pool name displayed if the AUTODS function is associated with an AUTOPOOL SMSPOOL= function (1-8 characters)
AD_TRKSA=*nnnnnnnnnn*	contains the number of tracks allocated (0-2147483647)
AD_TRKSF=*nnnnnnnnnn*	contains the number of tracks unused by the data set on the volume (0-2147483647) For VSAM data sets, the number of tracks unused is calculated from the high-allocated RBA and high-used RBA values for each volume. These values are obtained from the catalog.
AD_TRKSU=*nnnnnnnnnn*	contains the number of tracks used by the data set on the volume (0-2147483647) For VSAM data sets, the number of tracks used is calculated from the starting RBA and high-used RBA values for each volume. These values are obtained from the catalog.
AD_VOL=xxxxxx	specifies the volume number (1-6 characters)
AD_VOLSQ=*nnnnn*	specifies the volume sequence number for the data set (1-32767)
AD_XDATE=*xxxxxxxxxx*	specifies the expiration date for the data set in yyyy/mm/dd format
AUTOLEV=*xxxxxxxx*	contains an 8-characters literal AUTOLEV*x*, where *x* is a number indicating the current automation level for the resource being automated AUTOLEV0 indicates that multiple levels are not being used. AUTOLEV1 indicates the first level of a multiple level request. For multiple level automation this field must be used either in this FLST/RLST member or contained within event text of any event issued and then referenced in the AutoOPERATOR Rule that looks for the event.
SOLUTION=*xxxxxxxx*	contains the solution value from the originating AUTOVOL command (1-8 characters) It is recommended that each SET statement in the AUTOVOL function include a unique solution value. This solution value can then be used on the AUTOVOL console command to invoke the solution. For more information, see “SOLUTION Parameter” on page 4-2.

AUTODS Variables for Skeleton Tailoring

Variables created by the AUTODS function for use in skeleton JCL are shown in Table 4-7. These variables can be used in skeleton JCL when it is submitted using the ACT_JOB keyword in the AUTODS function. Other functions create a slightly different set of variables.

Table 4-7 AUTODS Variables for Skeleton Tailoring

Variable	Description
&VOLSER.*n*	contains the volume serials generated by the AUTODS function that should be included in the skeleton JCL
&DSN.*n*	contains a selected data set from AUTODS that should be included in job
&RESTYPE.*n*	indicates the resource type All array entries with the same value will contain one of the following values: pool, group, SMS pool–indicate AUTOVOL pool=/group=/and so on appl–indicates AUTOVOL APPL= was used volume–indicates AUTOVOL VOL= was used poolvol–indicates AUTODS POOL=, VOL= was used
&FUNC.*n*	contains number, will be same a &DSN.*n* &FUNC.*n* contains the AUTO function causing the job submission, in this case AUTODS. Created as stem, all entries will be the same value. This can be used as documentation in the skeleton JCL.
&SOL.*n*	contains the same number as &DSN.*n* &SOL.*n* contains a solution causing the job submission. All entries will be the same value. This can be used as documentation in the skeleton JCL.
&APPL.*n*, &POOL.*n,* &SMSPOOL.*n*, or &GROUP.*n*	resource, depending on which was used

AUTODS Function Usage Notes

The SET result group will contain data set information for each resource requested. The requested resource can be a pool, group, SMS pool, volume, or application. The SET result group will contain data set information for data sets within the resource that matched a RLST SET statement.

Sorting this list, for example by allocated tracks, and issuing events or submitting jobs causes actions on the specified number of largest allocated data sets within the requested resource.

AUTOAPPL Function – Application Automation

The AUTOAPPL function is used to initiate automation for applications. Application automation provides the capability to interrogate space or other pool attributes and initiate actions when one or any number of thresholds are exceeded. Actions include:

- submitting a job using a skeleton tailoring facility, similar to ISPF tailoring, to perform a batch process on the pool
- issuing events using the Enterprise Storage Automation component to communicate with AutoOPERATOR, allows all the actions available in AutoOPERATOR to be initiated in response to a pool condition.

Tip: SG-Control issues a system event when application usage exceeds certain thresholds. This event could be used in place of the AUTOAPPL function if you only want to take action at the time a threshold is exceeded. An AutoOPERATOR Rule on the SG-Control system event could invoke the same solution as any Rule that would fire on AUTOAPPL output events.

In conjunction with events being issued, AutoOPERATOR Rules can be used to initiate data set solutions for data sets in the application. These solutions can be taken step-by-step, checking the pool utilization at the end of each step.

The AUTOAPPL FLST/RLST statements are used to detect the situations or thresholds and specify the event or job actions to take. A number of application fields can be interrogated to determine when actions should be initiated.

Unlike other MAINVIEW SRM functions, this function is initiated through user request rather than during some other process or event. The AUTOAPPL console command can be used to initiate pool automation. This console command can be issued by time-driven Rules within AutoOPERATOR or through any mechanism that can issue console commands.

Note: AUTOVOL is not available from AUTOAPPL. There is no relationship between a application and a volume or volumes at this time.

Table 4-8 Filter List Parameters for AUTOAPPL (Part 1 of 2)

Parameter	Definition
AA_AMODE=*xxxx*	specifies the application mode Valid values are: • MON–Tracks space allocations and deallocations as they occur allowing up-to-the-minute analysis of DASD space usage. • WARN–A message is generated if the current allocation exceeds the budget amount. • REJ–If the current allocation exceeds the budget amount, the allocation will be rejected.
AA_APPL=*xxxxxxxxxxx*	specifies the SG-Control application name up to 50 characters If there is a blank in the SG-Control application name, enclose the application name in quotes.
AA_ASTAT=*xxxx*	specifies the application status Valid values are: • MDEL–The application has been manually flagged for deletion. The next time that the database is copied, this application will be deleted. • DEL–This application has been automatically flagged for deletion. This application was created, but never updated. Since no activity has taken place in the application, it will be deleted the next time the database is copied. • ACTV–This is a currently active application.
AA_CDATE=*xxxxxxxxxx*	indicates the 10- character date the application was created in the SG-Control database in yyyy/mm/dd format
AA_HSMC=*nnnnnnnnnn*	specifies the total amount of space allocated on DASD for HSM data sets assigned to this application (0-9223372036854775807) HSM tracking must be activated for the application in SG-Control for this field to have a value.
AA_HSMH=*nnnnnnnnnn*	indicates the largest amount of space allocated on DASD for HSM data sets assigned to this application (0-9223372036854775807) HSM tracking must be activated for the application in SG-Control for this field to have a value.
AA_KHSM=*Y/N*	indicates if HSM data sets are tracked for this application HSM tracking must be activated for the application in SG-Control for this field to have a value.
AA_KTEMP=*Y/N*	indicates if temporary data sets are tracked for this application
AA_KVSAM=*Y/N*	indicates if VSAM data sets are tracked for this application
AA_LDATE=*xxxxxxxxxx*	contains the date of the last allocation that resulted in one of the application fields being updated in yyyy/mm/dd format
AA_PERMC=*nnnnnnnnnn*	indicates the total amount of space allocated on DASD for permanent data sets assigned to this application (0-9223372036854775807)
AA_PERMH=*nnnnnnnnnn*	indicates the largest amount of space allocated on DASD for permanent data sets assigned to this application (0-9223372036854775807)
AA_PERMM=*nnnnnnnnnn*	indicates the maximum amount of space allowed for permanent data sets assigned to this application (0-9223372036854775807)
AA_PERMP=*nnn*	percentage of the permanent data set budget currently being used (0-100)

Table 4-8 Filter List Parameters for AUTOAPPL (Part 2 of 2)

Parameter	Definition
AA_PHSM=*Y/N*	indicates if HSM data set allocations are included as part of the permanent data set allocations
AA_PTEMP=*Y/N*	indicates if temporary data set allocations are included as part of the permanent data set allocations
AA_PVSAM=*Y/N*	indicates if VSAM data set allocations are included as part of the permanent data set allocations
AA_TEMPC=*nnnnnnnnn*	indicates the total amount of space allocated on DASD for temporary data sets assigned to this application (0-9223372036854775807)
AA_TEMPH=*nnnnnnnnnn*	indicates the largest amount of space allocated on DASD for temporary data sets assigned to this application (0-9223372036854775807)
AA_TEMPM=*nnnnnnnnnn*	specifies the maximum amount of space allowed for temporary data sets assigned to this application (0-9223372036854775807)
AA_TEMPP=*nnn*	percentage of the temporary data set budget currently being used (0-100)
AA_UFLD*n-xxxxxxxxxx*	1–3 SG-Control-defined fields AA_UFLD1 can contain up to 8 characters AA_UFLD2 and 3 can contain up to 10 characters
AA_UNAME=*xxxxxxxxxx*	contains an SG-Control-defined name up to 20 characters If there is a blank in the SG-Control user name, enclose the user name in quotes.
AA_VLCNT=*nnnnnnnnnn*	specifies the number of volumes that contain at least one data set included in the application's allocation amounts (0-2147483647)
AA_VSAMC=*nnnnnnnnnn*	indicates the total amount of space allocated on DASD for VSAM data sets assigned to this application (0-9223372036854775807)
AA_VSAMH=*nnnnnnnnnn*	indicates the largest amount of space allocated on DASD for VSAM data sets assigned to this application (0-9223372036854775807)
AA_VSAMM=*nnnnnnnnnn*	indicates the maximum amount of space allowed for VSAM data sets assigned to this application (0-9223372036854775807)
AA_VSAMP=*nnn*	percentage of the VSAM data set budget currently being used (0-100)
AA_WTHRS=*nnn*	specifies a percentage of the budget that can be used by the application before a warning message is issued (0-100)
AUTOLEV=*xxxxxxxx*	contains an 8-character literal AUTOLEV*x*, where *x* is a number indicating the current automation level for the resource being automated AUTOLEV0 indicates that multiple levels are not being used. AUTOLEV1 indicates the first level of a multiple level request. For multiple level automation this field must be used either in this FLST/RLST member or contained within event text of any event issued and then referenced in the AutoOPERATOR Rule that looks for the event.
SOLUTION=*xxxxxxxx*	contains the solution value from the originating AUTOAPPL command (1-8 characters) It is recommended that each SET statement in the AUTOAPPL function include a unique solution value. This solution value can then be used on the AUTOAPPL console command to invoke the solution. For more information, see “SOLUTION Parameter” on page 4-2.

AUTOAPPL Variables for Skeleton Tailoring

Variables created by the AUTOAPPL function for use in skeleton JCL are shown in Table 4-9. These variables can be used in skeleton JCL when it is submitted using the ACT_JOB keyword in the AUTOAPPL function. Other functions create a slightly different set of variables.

Table 4-9 AUTODS Variables for Skeleton Tailoring

Variable	Description
&APPL.*n*	indicates a selected application name that should be included in the job
&RESTYPE.*n*	indicates the resource type; will always be APPL
&FUNC.*n*	contains number, will be same as &APPL.*n* &FUNC.*n* contains the AUTO function causing the job submission, in this case AUTOAPPL. Created as stem, all entries will be the same value. This can be used as documentation in the skeleton JCL.
&SOL.*n*	contains the same number as &APPLN.*n* &SOL.*n* contains a solution causing the job submission. All entries will be the same value. This can be used as documentation in the skeleton JCL.

AUTOAPPL Function Usage Notes

The SET result group will contain application space information for each application in the automation request. Sorting this information, using SORT= in RLST, causes the space information for the applications in the request to be sorted. Any ACT_COUNT or ACT_SUM_LIM values, used to restrict the number of actions taken, will apply to the group. For example, the follow request is made:

```
SVOS AUTOAPPL Pool=(ApplA,ApplB,ApplC)
```

Space information for ApplA, ApplB, ApplC is obtained and run through FLST. The SET result group will include the applications from this list that matched a RLST SET statement. Sorting this list, for example by percent of budget, and issuing events or submitting jobs would cause actions on the specified number of top utilized applications in the request. Similar to AUTOPOOL and AUTOVOL, the volumes in the request determine the result group.

For AUTOAPPL requests that only specify a single application name, the SORT= RLST keyword and ACT_SUM_LIM keywords have no affect.

Job Submission and Skeleton Tailoring

There are two ways to cause a job to be submitted: ACT_JOB keyword, which submits one job containing all resource elements in the SET result group, and by an AutoOPERATOR Rule, invoking the MAINVIEW SRM CORSR005 REXX EXEC in response to an event. An explanation and diagrams of each method of job submission follows.

Job Submission Using ACT_JOB

When you use the ACT_JOB keyword, MAINVIEW SRM uses the Tailoring and Submit facilities of AutoOPERATOR. The value in this keyword identifies the JCL skeleton member to use. The member name must contain skeleton JCL and must be in a data set allocated in the SYSPROC DD statement in the AutoOPERATOR started task JCL. This JCL is tailored and submitted *once* for the entire SET result group of records. Unlike the ACT_EVENTID action keyword, which issues an event for each record in the SET result group, ACT_JOB causes the submission of just a single job that includes all SET result records.

When you use the AutoOPERATOR Skeleton Tailoring Facility, JCL can contain variables for data replacement. These variables can be any valid, defined AutoOPERATOR variable. Stem variables (tables) are also supported. This allows the use of directive statements like)DO to cause skeleton lines to be repeated a number of times, once for each stem value (table entry). In addition, when AutoOPERATOR Skeleton Tailoring is invoked from MAINVIEW SRM, additional variables and stem variables will be available for use in the skeleton JCL.

Job submission using ACT_JOB causes MAINVIEW SRM to create stem variables in AutoOPERATOR for each record in the FLST/RLST result group, to hold the pool, volume, and data set names. A REXX EXEC distributed with MAINVIEW SRM is then scheduled to the AutoOPERATOR address space to initiate the AutoOPERATOR Skeleton Tailoring Facility for the member name specified in ACT_JOB. The skeleton JCL can make use of the stem variables created by the MAINVIEW SRM function. As all records from the SET result group are passed in stem variables, the skeleton should make use of the)DO statement to ensure all of the stem entries are processed.

Each AUTO function creates a slightly different set of stem variables as each function processes different types of resources. See each AUTO function explanation for the list of variables. Examples of skeleton JCL can be found in a number of sample solutions documented in Chapter 6, “Implementing and Customizing Solutions.”

For information about skeleton tailoring, see the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

ACT_JOB and JOBNAMES

Enterprise Storage Automation keeps track of all jobs submitted for a resource or resources due to the ACT_JOB keyword. Enterprise Storage Automation uses AutoOPERATOR to determine when a submitted job ends. This is done through AutoOPERATOR Rules that look for MVS or JES messages issued when a job ends. These job-end Rules are distributed in Rule Set RULSRS01 and must be enabled if ACT_JOB is used. These job-end Rules invoke an Enterprise Storage Automation-supplied REXX EXEC that notifies Enterprise Storage Automation of the job end.

The job-end Rules filter on the JOBNAME of the submitted job. The job-end messages that cause this Rule to fire occur many times on an OS/390 system. To prevent AutoOPERATOR from scheduling this Enterprise Storage Automation REXX EXEC every time a job ends, a filter must be used. This requires one of the following implementations:

- All jobs submitted with the ACT_JOB keyword have a common prefix in the jobname. This prefix can then be filtered on in the job-end Rules. The prefix should be unique to ACT_JOB-submitted jobs to prevent execution of the job-end REXX EXEC for jobs not submitted through ACT_JOB. The job-end REXX EXEC only notifies Enterprise Storage Automation for jobs submitted with ACT_JOB; it does not cause a problem for the Rule to fire and the EXEC to run for a job not submitted by automation.

- Modify RULSRS01 to include a filter for each jobname submitted. This allows any jobname to be submitted using ACT_JOB; however, for each jobname a additional filter in the RULSRS01 Rules must be added.

To modify job-end AutoOPERATOR Rules:

Step 1 From your AutoOPERATOR primary menu, select **Basic and Advanced Automation**.

Step 2 Select **Display/Modify Rules and Rule Sets**.

Step 3 Verify that Rule Set RULSRS01 is enabled.

Step 4 Select the RULSRS01 Rule Set, placing you into the Rule Set Overview screen.

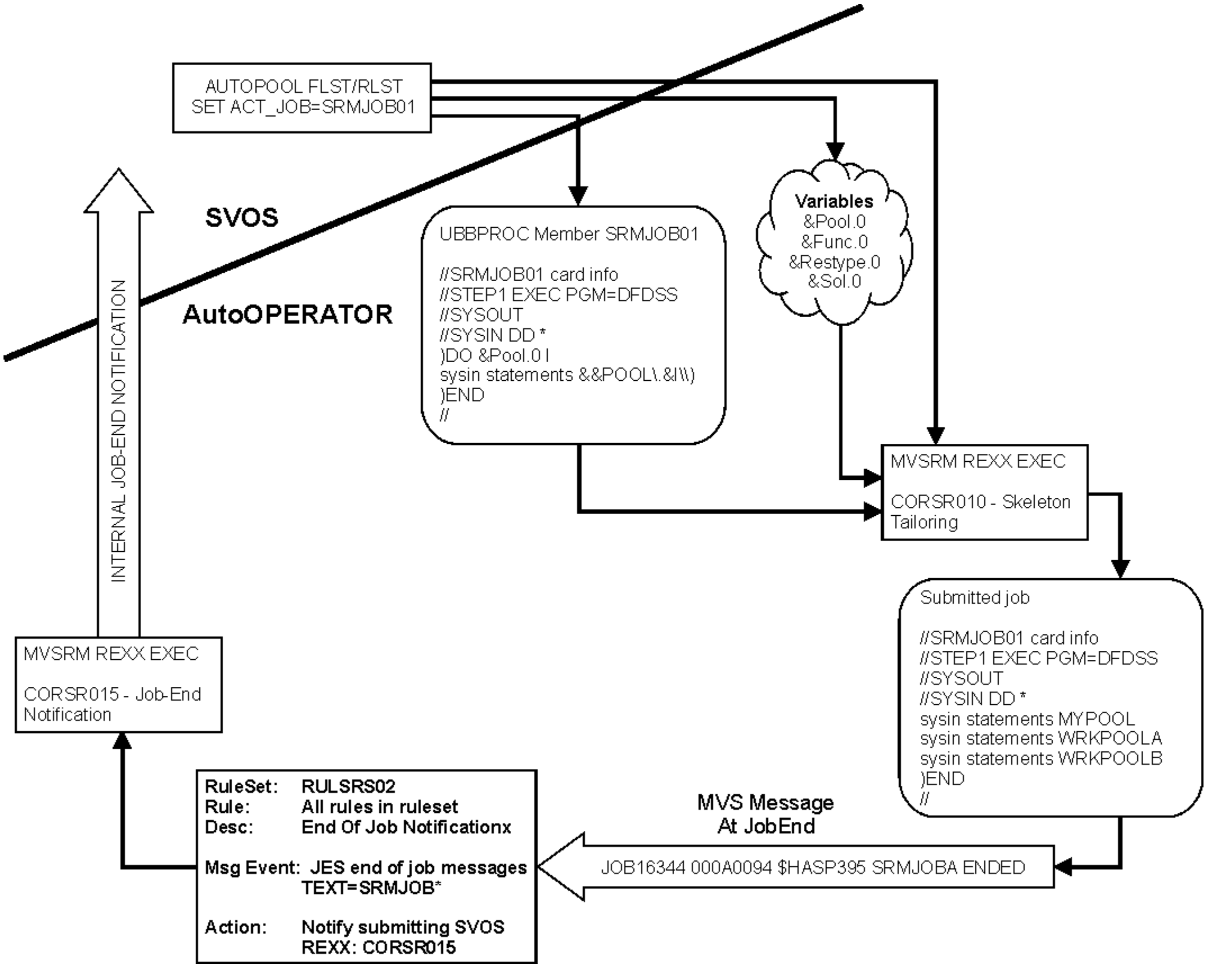
Step 5 Select the SRS01001 Rule ID; then enter SV (Variable Dependencies) on the command line. In the Selection Criteria panel you will see variable WORD2 being checked for a variable-value of SRM*. The SRM* value can be changed to a common prefix used in ACT_JOB jobnames, or additional lines may be added, with OR in the AND/OR column, to add checks for other jobnames or jobname prefix values.

Step 6 Repeat this process for each SRS01*nnn* Rule ID in the RULSRS01 Rule Set.

Step 7 Remember to save the changes when you exit the Rule Set Overview.

Figure 4-1 shows the job submission process when ACT_JOB is used.

Figure 4-1 Job Submission with ACT_JOB Diagram



```
AUTOPOOL FLST/RLST
SET ACT_JOB=SRMJOB01
```

```
UBBPROC Member SRMJOB01

//SRMJOB01 card info
//STEP1 EXEC PGM=DFDSS
//SYSOUT
//SYSIN DD *
)DO &Pool.0 I
sysin statements &&POOL\.&I\\)
)END
//
```

```
MVSRM REXX EXEC

CORSR010 - Skeleton
Tailoring
```

```
Submitted job

//SRMJOB01 card info
//STEP1 EXEC PGM=DFDSS
//SYSOUT
//SYSIN DD *
sysin statements MYPOOL
sysin statements WRKPOOLA
sysin statements WRKPOOLB
)END
//
```

```
JOB16344 000A0094 $HASP395 SRMJOBA ENDED
```

```
RuleSet:    RULSRS02
Rule:       All rules in ruleset
Desc:       End Of Job Notificationx

Msg Event:  JES end of job messages
            TEXT=SRMJOB*

Action:     Notify submitting SVOS
            REXX: CORSR015
```

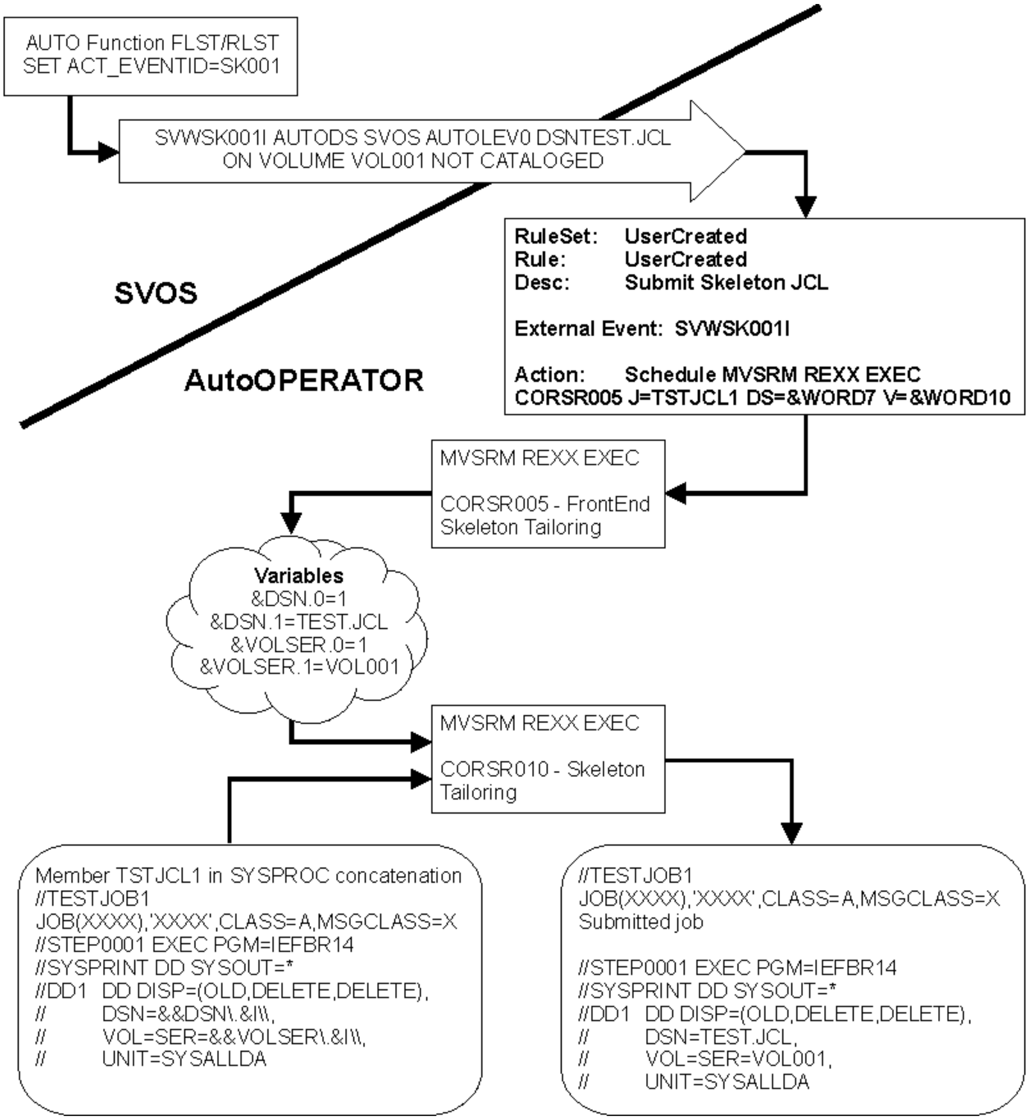
```
MVSRM REXX EXEC

CORSR015 - Job-End
Notification
```

Job Submission from Events

You can submit a job using a console command, which invokes REXX EXEC CORSR005, or you can create an event and an AutoOPERATOR Rule definition that passes parameters to the REXX EXEC CORSR005. CORSR005 processes the parameters that are passed as variables and invoke AutoOPERATOR Skeleton Tailoring to build the JCL and submit the batch job. The parameters passed are converted to variables placed in array format using stem variable names. This allows the same skeleton JCL used with the ACT_JOB RLST keyword to be used when invoked from an event. For detailed information about this method of job submission, see “CORSR005 – Job Submission Using Console Commands” on page B-2.

Figure 4-2 shows the job submission process when using REXX EXEC CORSR005.

Figure 4-2 Job Submission from an Event Diagram



```
SVWSK001I AUTODS SVOS AUTOLEV0 DSNTEST.JCL
ON VOLUME VOL001 NOT CATALOGED
```

```
RuleSet:   UserCreated
Rule:      UserCreated
Desc:      Submit Skeleton JCL

External Event:  SVWSK001I

Action:    Schedule MVSRM REXX EXEC
CORSR005 J=TSTJCL1 DS=&WORD7 V=&WORD10
```

```
&DSN.0=1
&DSN.1=TEST.JCL
&VOLSER.0=1
&VOLSER.1=VOL001
```

```
Member TSTJCL1 in SYSPROC concatenation
//TESTJOB1
JOB(XXXX),'XXXX',CLASS=A,MSGCLASS=X
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1   DD DISP=(OLD,DELETE,DELETE),
//         DSN=&&DSN\.&I\,
//         VOL=SER=&&VOLSER\.&I\,
//         UNIT=SYSALLDA
```

```
//TESTJOB1
JOB(XXXX),'XXXX',CLASS=A,MSGCLASS=X
Submitted job

//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1   DD DISP=(OLD,DELETE,DELETE),
//         DSN=TEST.JCL,
//         VOL=SER=VOL001,
//         UNIT=SYSALLDA
```

Sysplex Considerations When Using ACT_JOB

If you plan to implement ACT_JOB in an environment where the job could be submitted on one system within the sysplex and executed on another, either by taking advantage of Parallel Sysplex functionality or by use of a route statement in the JCL, the following criteria must be met.

- An AutoOPERATOR address space must be executing on the destination system.

- The subsystem IDs of the AutoOPERATOR address spaces running within the sysplex must be unique.

- The AutoOPERATOR subsystem on the destination system must be in active communication with the AutoOPERATOR on the system that submitted the job.

- Rule Set RULSRS01 must be customized and active within the AutoOPERATOR address space on both systems (the system that submitted the job and the system where the job executes).

- The MAINVIEW SRM REXX EXECs must be in the SYSPROC concatenation of AutoOPERATOR address spaces of both systems (the system that submitted the job and the system where the job executes). For more information about data set concatenation and customization, see the *MAINVIEW SRM Customization Guide*.

Multi-Level Automation (MLA) Processing

Multi-level automation (MLA) refers to the process of exceeding a threshold, performing an action, checking the threshold again, and, if still exceeded, performing another action, and so on, until the threshold is no longer exceeded or until there are no other actions to take. This repetitive process can be performed up to 10 times, taking 10 different actions along the way, or until there are no more actions defined.

To use multi-level automation, you must:

Step 1 Code MLA=Y on the AUTOPOOL, AUTOVOL, or AUTOAPPL console request. MLA=Y sets the AUTOLEV to 1, indicating the first step in the MLA request. On completion of each submitted job, the AUTOLEV is incremented and the request is redriven for the resource(s).

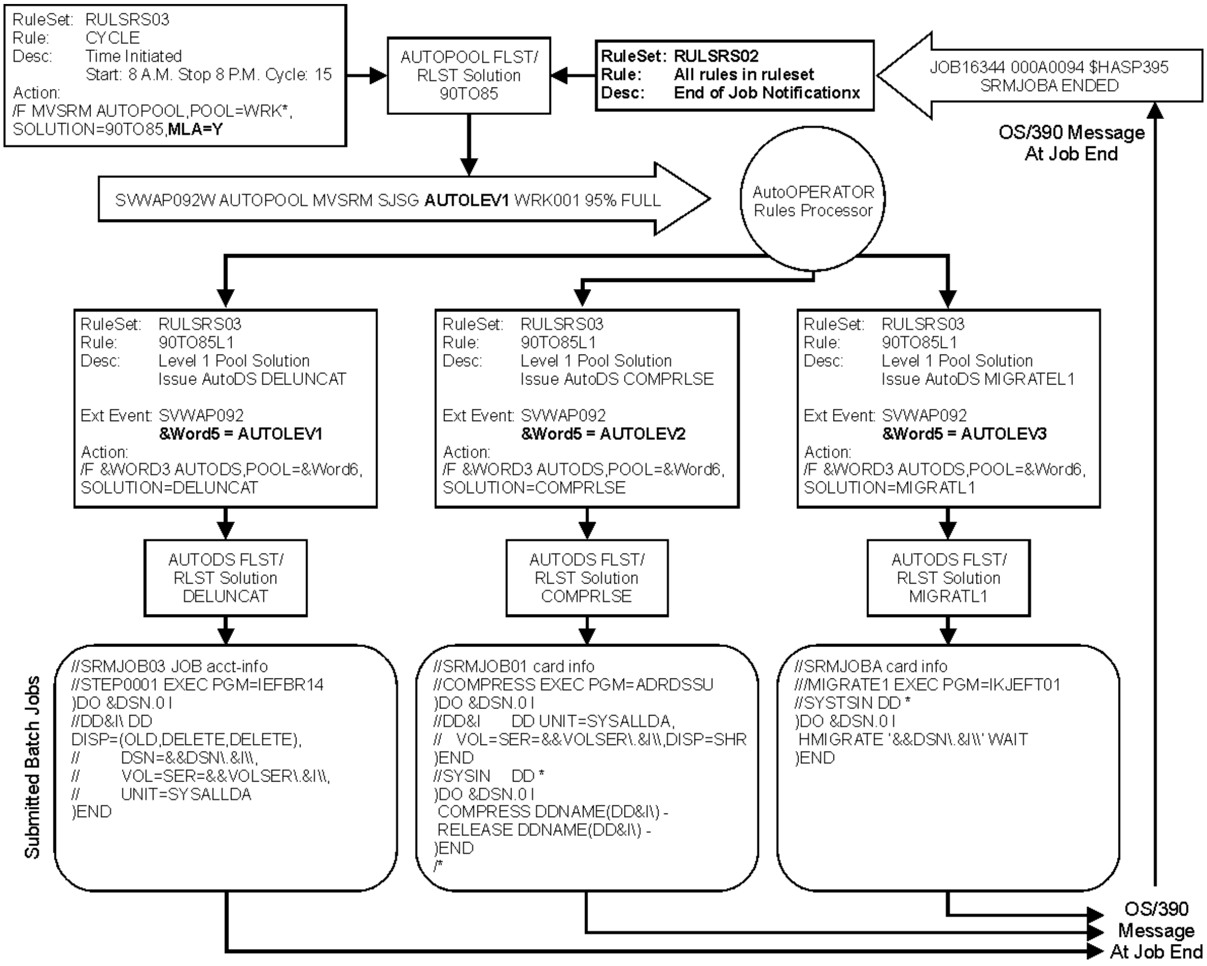
Step 2 Filter on the AUTOLEV value. Coding MLA=Y on the request causes the AUTOLEV value to be maintained and the request to be redriven on completion of each solution. Filtering on the AUTOLEV value controls the actions taken at each level. You can filter on the AUTOLEV value in either the RLST statements for the solution or by placing AUTOLEV in the event message text and using AUTOLEV in AutoOPERATOR Rules.

Step 3 Make sure the last action of each step, or automation level, is a solution using the ACT_JOB keyword. The initial AUTO request is redriven only when a job submitted through ACT_JOB keyword completes. The request is not redriven when a solution results in an event being issued. This is not to say that events cannot be issued in an MLA solution; however, an AutoOPERATOR Rule should be set up to fire on that event and issue another request on the resource, continuing automation processing. This use of events during MLA processing is described further in this section and is also used in sample solutions in Chapter 6, “Implementing and Customizing Solutions.”

Step 4 Enable an additional AutoOPERATOR Rule to notify automation processing when a submitted job ends, as shown in Figure 4-3**.** Any time the ACT_JOB keyword is used the AutoOPERATOR Rules in Rule Set RULSRS01 must be enabled. This filter requirement on the jobname is discussed in more detail in in this section. For more info see “Skeleton Tailoring and Job Submission” on page 3-7

All of these steps can be seen in Figure 4-3.

Figure 4-3 AUTOPOOL Multi-Level Example



```
RuleSet:  RULSRS03
Rule:     CYCLE
Desc:     Time Initiated
          Start: 8 A.M. Stop 8 P.M. Cycle: 15
Action:
/F MVSRM AUTOPOOL,POOL=WRK*,
SOLUTION=90TO85,MLA=Y
```

```
RuleSet: RULSRS02
Rule:    All rules in ruleset
Desc:    End of Job Notificationx
```

```
JOB16344 000A0094 $HASP395
SRMJOBA ENDED
```

```
SVWAP092W AUTOPOOL MVSRM SJSG AUTOLEV1 WRK001 95% FULL
```

```
RuleSet:  RULSRS03
Rule:     90TO85L1
Desc:     Level 1 Pool Solution
          Issue AutoDS DELUNCAT

Ext Event SVWAP092
          &Word5 = AUTOLEV1
Action:
/F &WORD3 AUTODS,POOL=&Word6,
SOLUTION=DELUNCAT
```

```
RuleSet:  RULSRS03
Rule:     90TO85L1
Desc:     Level 1 Pool Solution
          Issue AutoDS COMPRLSE

Ext Event SVWAP092
          &Word5 = AUTOLEV2
Action:
/F &WORD3 AUTODS,POOL=&Word6,
SOLUTION=COMPRLSE
```

```
RuleSet:  RULSRS03
Rule:     90TO85L1
Desc:     Level 1 Pool Solution
          Issue AutoDS MIGRATEL1

Ext Event SVWAP092
          &Word5 = AUTOLEV3
Action:
/F &WORD3 AUTODS,POOL=&Word6,
SOLUTION=MIGRATL1
```

```
//SRMJOB03 JOB acct-info
//STEP0001 EXEC PGM=IEFBR14
)DO &DSN.0 I
//DD&I\ DD
DISP=(OLD,DELETE,DELETE),
//        DSN=&&DSN\.&I\\,
//        VOL=SER=&&VOLSER\.&I\\,
//        UNIT=SYSALLDA
)END
```

```
//SRMJOB01 card info
//COMPRESS EXEC PGM=ADRDSSU
)DO &DSN.0 I
//DD&I       DD UNIT=SYSALLDA,
//   VOL=SER=&&VOLSER\.&I\\,DISP=SHR
)END
//SYSIN    DD *
)DO &DSN.0 I
 COMPRESS DDNAME(DD&I\) -
 RELEASE DDNAME(DD&I\) -
)END
/*
```

```
//SRMJOBA card info
///MIGRATE1 EXEC PGM=IKJEFT01
//SYSTSIN DD *
)DO &DSN.0 I
 HMIGRATE '&&DSN\.&I\\' WAIT
)END
```

Automation Level

MLA is controlled by use of the automation level value. The automation level indicates the level, or step, in the action sequence being performed on the resource. With the initial MLA=Y request, the automation level is set to one. The MLA indicator as seen on the AUTO view for the resource is set to Y and the Repeat Function and Repeat Solution as seen in the AUTO view are set to the current request. As each action completes, the automation level is incremented and the initial AUTO request, in Repeat Function and Repeat Solution, is redriven for the resource. The automation level value is available in AUTO functions RLST INC/EXC statements and can also be included in event message texts for filtering in AutoOPERATOR Rules. The automation level is used to control the actions taken at each level, as shown in the Figure 4-3.

When MLA is not being used the automation level is zero. MLA processing is stopped when either an event is generated or when the RLST condition filters are no longer met:

- When an *event is generated*, the automation level remains at the level causing the event to be generated, the MLA indicator remains set to Y, and the repeated function and solution, as seen on the AUTO view, remain for the request. If another request that does not specify the MLA= parameter is received for the resource, it is considered in the same automation level. If another request that specifies MLA=N is received, the automation level is set back to zero, MLA is set to N, and the repeat function and solution values are cleared, as can be seen in the AUTO view. In the Pool Space Management Automation solution, which is diagramed in Figure 6-5, MLA processing stops when the event is issued at AUTOLEV4. As there are no AutoOPERATOR Rules to fire on AUTOLEV4 no other actions are initiated for the resource.

- When the *RLST condition filters are no longer met*, the automation level is set to zero, MLA is set to N, and the repeat function and solution are cleared, as can be seen in the AUTO view.

MLA Parameter

The following summarizes when each of the possible MLA specifications are used:

- **MLA=Y** is used to start MLA processing. The AUTO function and solution requested on the AUTO command is redriven at the end of each action taken, until one of the two conditions described above are encountered.

- **MLA=N** can be used to ensure that multi-level automation does not take place for a request. This specification can be used to set the AUTOLEV to AUTOLEV0, set MLA to N, and clear the Repeat Function and Solution values as seen on the AUTO view in situations where an MLA request did not run through the condition being resolved.

- **MLA not specified**. If no MLA parameter is specified, the request runs with the current MLA attributes of the resource. If the current MLA value of the resource is N, the request runs with no MLA processing. If the current MLA value of the resource is Y, the request runs as part of the current MLA process with the current AUTOLEV value for the resource. This can be used when issuing an event from an MLA solution that fires an AutoOPERATOR Rule to issue another AUTO request on the resource. This subsequent AUTO request should not specify the MLA parameter, as this request will be part of the current MLA automation level for the resource. This can be seen in Figure 4-3, where event AP092 is generated and causes AutoOPERATOR Rule SRS03002 to fire, which issues the AUTODS DELUNCAT solution on the pool. This AUTODS command does not specify MLA, and, therefore; runs as part of the current automation level for the resource. In the example, this is done on the first automation level for the resource. When the job submitted by the AUTODS DELUNCAT solution ends, the automation level is incremented and the Repeat Function and Repeat Solution are redriven, in this case AUTOPOOL function with solution 90TO85.

Chapter 5 Using AUTO Commands

Storage automation is initiated through a console command issued to the MAINVIEW SRM SVOS address space. You can issue the console command manually, through AutoOPERATOR Rules, or through any other mechanism that allows the issuance of console commands (MAINVIEW Alarm, MAINVIEW API within AutoOPERATOR, and so on).

To use AUTO commands

- Enterprise Storage Automation must be active for the AUTO console commands to be accepted

- the requested AUTO function must be active

- for AUTOAPPL, the SG-Control component must be licensed and active

Listed below are the console commands available in Enterprise Storage Automation. They are described in greater detail in the remaining pages of this book.

Overview

The AUTO commands are the interface to the AUTO functions, a real-time collection facility that also includes user controls for job submission and event generation. As the AUTO commands are the interface to the AUTO functions, an understanding of the AUTO functions, and in particular the SOLUTION= keyword, is needed to properly use the commands. When an AUTO command is issued, the AUTO functions data collection processes are started. These process can include scanning VTOCs and obtaining data set-level information across the resources indicated in the command. Depending on the number of volumes and/or data sets associated with a resource, this collection process could be time and CPU intensive. This situation can be exacerbated when volumes are included in multiple applications.

You can use the AUTO_MXTSK and VSCAN_MXTSK keywords to reflect the relative importance of automation processing. These keywords specify a number of tasks that will be used in automation processing. Automation processing, using AUTO functions, is designed to spread its processing amongst any number of tasks. As this number is increased, more tasks are used, improving elapsed time of the automation collection process at the cost of CPU and storage utilization in the SVOS address space. The minimum value for these keywords is two (2).

Information on each resource included in an AUTO command, its status, number of requests received including the resource, and other information is maintained by Enterprise Storage Automation and displayed in the AUTO view within the MAINVIEW dialog.

The number of events issued, jobs submitted, the last function requested, and other information for each resource is also included to monitor the effects of the AUTO commands and the processing of the associated functions.

A resource cannot be manipulated concurrently by different requests during automation processing. This prevents automation loops, where a solution takes longer to complete than the interval at which the resource is being monitored. This also ensures that the effects of a solution are realized before any other solutions are started. A resources status is set to active when an AUTO request specifying the resource is received. The status remains active until one of the following occurs:

- If a job is submitted for the resource, the resource status is set to Waiting. This status remains until the submitted job ends.

- If an event is issued for the resource, the resource becomes inactive at the time the event is issued.

- If the resource is not processed by the requested solution, meaning it does not match the solution SET statements INC/EXC conditions, then the resource becomes inactive immediately after the AUTO function collection process completes.

Requests for a resource that is already active will be rejected and a message indicating this will be issued to the SVOS job log.

AUTOPOOL Command

The AUTOPOOL command is used to initiate an automation solution on a list of pools, storage groups, SMS pools, or applications. Only one of these resource types may be specified in a single request.

The AUTOPOOL command has the following possible keywords:

Keyword **Description**

POOL= specifies a list of MAINVIEW SRM pool names or pool name masks on which to perform the AUTOPOOL function

The list can contain any number of names or mask values, limited only by the space available on the console command entry. If a mask is used, any defined pool matching the mask value is automated. To specify a list of SMS storage groups, see GROUP=. To specify a list of SMS Pools, see SMSPOOL=. Either POOL=, GROUP=, or SMSPOOL= must be specified to indicate the resources to automate. Only one of these keywords may be specified on the command.

GROUP= specifies a list of SMS storage group names or storage group name masks

SMSPOOL= specifies a list of MAINVIEW SRM-defined SMS pool names or SMS pool name masks

SOLUTION= specifies a value to use as the solution for the request

This optional value is used in the **SOLUTION** field in the pool record, where it can then be inspected by the AUTOPOOL function FLST/RLST members. This value associates the request with a particular SET statement in the AUTOPOOL RLST member. For more information, see “SOLUTION Parameter” on page 4-2.

If this keyword is not specified, the **SOLUTION** field in the pool record will contain blanks. No editing of this input command field is performed. Any value, up to 8 characters, can be specified. The field in the pool record contains the specification, padded on the right up to 8 bytes with spaces. It is recommended that every request specify a solution value. This solution value should match a value filtered in an AUTOPOOL RLST SET statement. Abbreviation: SOL=

MLA= specifies whether multi-level automation should be started for the specified list of pools/groups/SMS pools

Specifying Y in this field causes the automation level field of the pool to be set to AUTOLEV1, indicating the first level. Specifying MLA=N will set the automation level to zero. Not specifying the MLA keyword will cause the auto level to be unaffected. For more information about multi-level automation, see “Multi-Level Automation (MLA) Processing” on page 4-35.

AUTOPOOL Command Usage Notes

The pools specified on a request are processed in the same iteration of the AUTO function. These pools, or a subset of these pools, will make up the SET result group for the requested solution. Any SORT= or other keywords specified in the SET statement will occur on this list.

So, for example, if an RLST solution SET statement uses SORT= on % full and ACT_COUNT=5 to take action on the 5 highest utilized pools, it is performed on the highest utilized pools in the list specified on the AUTOPOOL POOL= command. If the list contains 5 or fewer pools, then the action will be taken on all the pools. For more information, see “Rule List SET Parameters” on page 4-5.

Specifying MLA=Y on the AUTOPOOL command initiates multi-level automation. This will cause the specified AUTOPOOL solution to be redriven for each pool that is acted upon though job submission. For more information, see “Multi-Level Automation (MLA) Processing” on page 4-35.

Example

This example shows how to use AUTOPOOL keywords.

```
/-SVOS AUTOPOOL,POOL=(DOUG*,POOL202), MLA=Y,SOLUTION=SPACE
/-SVOS AUTOPOOL,GROUP=(GRP001),SOLUTION=MYTEST
/-SVOS AUTOPOOL,SMSPOOL=(SMSP001),SOLUTION=POOLFRAG
```

AUTOVOL Command

The AUTOVOL command is used to initiate volume automation solutions against volumes in a pool, storage group, SMS pool, or on a list of individual volumes.

The AUTOVOL command has the following possible keywords:

Keyword **Description**

POOL= specifies a single MAINVIEW SRM pool name on which to perform the AUTOVOL function

This must be a single, fully qualified pool name, meaning no masking can be used. To specify an SMS storage group, see GROUP=. To specify an SMS Pool, see SMSPOOL=. To specify a list of volumes, see VOL=. Either POOL=, GROUP=, SMSPOOL=, or VOL= must be specified to indicate the resource to automate. Only one of these keywords may be specified in the command.

GROUP= specifies an SMS storage group name

SMSPOOL= specifies a MAINVIEW SRM-defined SMS pool name

VOL= specifies a list of volume serials or volume serial name masks on which to perform the AUTOVOL function

The list can contain any number of names or mask values, limited only by the space available on the console command entry. If a mask is used, any defined volume serial matching the mask value is automated

SOLUTION= specifies a value to use as the solution for the request

This optional value is used in the **SOLUTION** field in the volume record, where it can then be inspected by the AUTOVOL function FLST/RLST members. This value associates the request with a particular SET statement in the AUTOVOL RLST member. For more information, see “SOLUTION Parameter” on page 4-2.

If this keyword is not specified, the **SOLUTION** field in the volume record will contain blanks. No editing of this input command field is performed. Any value, up to 8 characters, can be specified. The field in the volume record contains the specification, padded on the right up to 8 bytes with spaces. It is recommended that every request specify a solution value. This solution value should match a value filtered in an AUTOVOL RLST SET statement. Abbreviation: SOL=

MLA= specifies whether multi-level automation should be started for the specified pool/group/SMS pool or volume list

Specifying Y in this field causes the automation level field of the pool to be set to AUTOLEV1, indicating the first level. Specifying MLA=N will set the automation level to zero. Not specifying the MLA keyword will cause the auto level to be unaffected. For more information about multi-level automation, see “Multi-Level Automation (MLA) Processing” on page 4-35.

AUTOVOL Command Usage Notes

AUTOVOL processes a list of volumes. The volumes are either in a pool or listed together on the AUTOVOL request. These volumes, or a subset of these volumes, will make up the SET result group manipulated by the SET keywords of AUTOVOL RLST. For more information, see “Rule List SET Parameters” on page 4-5.

Specifying MLA=Y on the AUTOVOL command initiates multi-level automation. This will cause the specified AUTOVOL solution to be redriven for each pool that is acted upon though job submission. For more information, see “Multi-Level Automation (MLA) Processing” on page 4-35.

Example

This example shows how to use AUTOVOL keywords.

```
/-SVOS AUTOVOL,POOL=DOUGPOOL, MLA=Y,SOLUTION=FREESPAC
/-SVOS AUTOVOL,VOL=(BAB*,WRK*,SYS001,MYVOL), SOLUTION=RLSESPAC
```

AUTODS Command

The AUTODS command is used to initiate data set solutions against data sets in a pool, storage group, SMS pool, application, or against data sets on a volume, or on a volume within a pool.

The AUTODS command has the following possible keywords:

Keyword **Description**

POOL= specifies a single MAINVIEW SRM pool name on which to perform the AUTODS function

This value must be a single, fully-qualified pool name, meaning no masking can be used. VOL= can be used in conjunction with POOL= to represent a volume in a pool.

This keyword cannot be specified with the GROUP=, SMSPOOL=, and APPL= keywords. Only one of these keywords may be specified on the command.

GROUP= specifies an SMS storage group name

SMSPOOL= specifies a MAINVIEW SRM-defined SMS pool name

VOL= specifies the volume serial number

VOL= can be specified by itself to initiate an AUTODS solution against a individual volume.

VOL= can be used with POOL= to indicate a volume in a pool. This specification is for multi-level automation usage, where the AUTODS solution is being initiated on multiple volumes in a pool, each on a separate AUTODS command. In this scenario, each request is associated with the pool. For more information, see “Multi-Level Automation (MLA) Processing” on page 4-35.

APPL= specifies an SG-Control application name

SOLUTION= specifies a value to use as the solution for the request

This optional value is used in the **SOLUTION** field in the data set record. This field can then be inspected in the AUTODS function FLST/RLST members. This value associates the request with a particular SET statement in the AUTODS RLST member. For more information, see “SOLUTION Parameter” on page 4-2.

If this keyword is not specified, the **SOLUTION** field in the data set record will contain blanks. No editing of this command input field is performed. Any value, up to 8 characters, can be specified. The field in the data set record contains the specification, padded on the right up to 8 bytes with spaces. It is recommended that every request specify a solution value. This solution value should match a value filtered in an AUTODS RLST SET statement. Abbreviation: SOL=

Note: MLA= can not be specified on this command.

AUTODS Command Usage Notes

AUTODS processes the data sets associated with the specified resource (pool, application, volume, and so on). These data sets, or a subset of these data sets, will make up the result group manipulated by the SET keywords of AUTODS RLST. For more information, see “Rule List SET Parameters” on page 4-5.

AUTODS requests cause collection to occur for all data sets in the indicated resource. Depending on the size of the resource, this collection could be CPU and elapsed time intensive.

Example

This example shows how to use AUTODS keywords

```
/-SVOS AUTODS,POOL=DOUGPOOL,SOLUTION=FREESPAC
/-SVOS AUTODS,VOL=BAB200,SOLUTION=RLSESPAC
```

AUTOAPPL Command

The AUTOAPPL command is used to initiate application-level solutions on a list of application names or application name masks. The following keywords are available for the AUTOAPPL command:

Keyword **Description**

APPL= specifies a list of MAINVIEW SRM SG-Control-defined application names or application name masks on which to perform the AUTOAPPL function

The list can contain any number of names or mask values, limited only by the space available on the console command entry. If a mask is used, any defined application name matching the mask value is automated. This is a required keyword. Application names containing spaces must be enclosed in apostrophes.

SOLUTION= specifies a value to use as the solution for the request

This optional value is used in the **SOLUTION** field in the application record. This field can then be inspected in the AUTOAPPL function FLST/RLST members. This value associates the request with a particular SET statement in the AUTOAPPL RLST member. For more information, see “SOLUTION Parameter” on page 4-2.

If not specified the **SOLUTION** field in the application record will contain blanks. No editing of this command input field is performed. Any value, up to 8 characters, can be specified. The field in the application record contains the specification, padded on the right up to 8 bytes with spaces. It is recommended that every request specify a solution value. This solution value should match a value filtered in an AUTOAPPL RLST SET statement. Abbreviation: SOL=

MLA= specifies whether multi-level automation should be started for the specified application list

Specifying Y in this field causes the automation level field of the application to be set to AUTOLEV1, indicating the first level. Specifying MLA=N will set the automation level to zero. Not specifying the MLA keyword will cause the auto level to be unaffected. For more information about multi-level automation, see “Multi-Level Automation (MLA) Processing” on page 4-35.

AUTOAPPL Command Usage Notes

The applications specified on a request are processed in the same iteration of the AUTO function. These applications, or a subset of these applications, will make up the SET result group for the requested solution. Any SORT= or other keywords specified in the SET statement will occur on this list.

So, for example, if an RLST solution SET statement uses SORT= on % full and ACT_COUNT=5 to take action on the 5 highest utilized applications, it is performed on the highest utilized applications in the list. If the list contains 5 or fewer applications, then the action will be taken on all the applications. For more information on SET keywords, see “Rule List SET Parameters” on page 4-5.

Specifying MLA=Y on the AUTOAPPL command initiates multi level automation. This will cause the specified AUTOAPPL solution to be redriven for each application that is acted upon though job submission. For more information, see “Multi-Level Automation (MLA) Processing” on page 4-35.

Example

This example shows how to use AUTOAPPL keywords.

```
/-SVOS AUTOAPPL,APPL=(DOUGAPPL,SYS*),MLA=Y,SOLUTION=SPACE
```

Example

This example shows the use of apostrophes when the application name contains spaces.

```
APPL=(SYS*,'DAVE WORK APPL')
```

JOB END Command

The JOB END command is issued to SVESA internally by a MAINVIEW SRM-provided REXX EXEC, which is initiated from AutoOPERATOR Rules that detect a job ending. These Rules are distributed in Rule Set RULSRS01 and must be active whenever a solution using the ACT_JOB keyword is requested. See the "Job Submission and Skeleton Tailoring" on page 4-29 for more information on job submission and detection of jobs ending.

The JOB END command can be used in situations where either the RULSRS01 Rule Set was not enabled at the time a job ended or the filters in the Rules did not correspond to the jobname of the submitted job. In these situations, the resource will remain in a Waiting status, as displayed in the AUTO view of the MAINVIEW dialog. While a resource is in a Waiting status, waiting for a job to end, no other AUTO commands will be accepted against the resource. The JOB END commands can be issued manually in these error situations to clear up the Waiting condition on the resource without having to cycle SVESA to clean up the condition.

The JOB END command specifies the AUTO function that caused the job submission and the resource name of the resource. It also contains the JOBEND positional parameter, indicating this is a notification of a job ending. The SOL= and MLA= parameters of the AUTO command are not needed on the JOB END command, and if specified are ignored.

If MLA is currently in progress for the resource, the JOB END command will cause the automation level to be incremented and the MLA-requested function and solution to be redriven for the resource. See "Multi-Level Automation (MLA) Processing" on page 4-35 for more information on MLA.

The following are forms of the JOB END command:

/F *mvsrmas*,SVOS AUTOPOOL POOL=DOUG,JOBEND

this could be used when pool DOUG is in a Waiting status.

/F *mvsrmas*,SVOS AUTOVOL VOL=WORK01,JOBEND

this can be used when volume WORK01 is in a Waiting status.

/F *mvsrmas*,SVOS AUTOAPPL APPL=*applname*,JOBEND

/F *mvsrmas*,SVOS AUTODS POOL=DOUG,JOBEND

This can also be used to notify SVESA that a job for pool DOUG has ended. Note that this command and the AUTOPOOL JOB END command above have the same result. BMC Software recommends that the AUTO command causing the job submission be used for the JOB END command, but this is not required.

Chapter 6 Implementing and Customizing Solutions

This chapter provides a complete description of the solutions distributed with Enterprise Storage Automation.

Distributed Solutions Overview

BMC Software distributed solutions can encompass FLST/ RLST statements, SMEVNT*xx* definitions, AutoOPERATOR Rules in Rule Sets, and skeleton JCL members. Some of the solutions may simply issue events, while others may use job submission and/or AutoOPERATOR Rules to perform their process.

The library members used in the distributed solutions should have been copied into your production libraries during the customization step of the install process. To review, the members can be found in the libraries noted in Table 6-1.

Table 6-1 Library Members

Element	Distributed to...	Copied to...
Function definitions FLST/RLST definitions	*?prefix*.BBPARM	*?prefix*.UBBPARM
REXX EXECs	*?prefix*.BBCLIB	AutoOPERATOR *?prefix*.UBBPROC
AutoOPERATOR Rules	*?prefix*.BBSAMP	AutoOPERATOR *?prefix*.UBBPARM
Skeleton JCL	*?prefix*.BBSAMP	AutoOPERATOR *?prefix*.UBBPROC

All Enterprise Storage Automation distributed Rule Sets are in a disabled status when AutoOPERATOR is started. You can cause these Rule Sets to be automatically enabled when AutoOPERATOR is started by adding them to the Rule Set parameter in AutoOPERATOR *?prefix*.UBBPARM (AAOPRM00). All Rules in the distributed Rule Sets are defined as enabled.

Because the definitions within the elements that constitute each solution vary, BMC Software suggests that you read through the supplied documentation for each solution, even if you do not plan to implement it. Doing so will not only give you a better understanding of the solution but it will also give you a better understanding of the automation process in general.

All solutions are coded to perform a specific task to attain a specific goal. Altering the elements within a solution, like the FLST/RLST statements, changes the outcome of the solution. These distributed solutions may not meet the goals as required by your installation. Therefore, before implementing any of the distributed solutions, review the FLST/RLST selection criteria and the SET keywords for values appropriate for your environment. Also check the content of the JCL statements for the batch jobs and the content of the AutoOPERATOR Rules.

There is an extreme amount of flexibility in the creation and implementation of solutions. When creating and implementing solutions, keep in mind that it may not be appropriate to run certain solutions simultaneously or during certain times of the day, depending on the operations of your installation.

You also must understand the goals you wish to achieve using the solutions and how to create or modify solutions to meet those goals. With that information, along with knowledge of the operation of your installation, you can devise a plan for the implementation of those solutions.

Distributed Automation Solutions

Some of the distributed solutions use the ACT_JOB keyword to submit skeleton JCL. To assist in familiarization, customization, and testing of these distributed solutions, the ACT_EVENTID parameter has also been coded in the RLST of these solutions, as a comment line. A matching event has been coded in the SMEVNT00 member. This ACT_EVENTID keyword can be used in place of the ACT_JOB keyword to cause event generation rather than job submission. Using the ACT_EVENTID keyword and the installation and activation of AutoOPERATOR Rule Set RULSRS02, you will be able to view the results of the solution in the AutoOPERATOR Journal display without any action being taken against any resource.

All Automated solutions that use the ACT_JOB parameter in the RLST statements must also have AutoOPERATOR Rule Set RULSRS01 installed and enabled. The Variable Dependencies of each Rule within the Rule Set must include variable values to match the job name used during automation. For more information, see “Job Submission and Skeleton Tailoring” on page 4-29.

The skeleton JCL provided contains a generic JOB card that needs to be customized for your environment. Remember, when updating this job card and specifying the jobname, the Rules in Rule Set RULSRS01 must be updated with the value. The control card statements and or DD statements have been commented out in the JCL. Using the JCL this way allows for variable substitution to take place in the skeleton JCL, the submission of the batch job, and allows the job to run to completion while no action is taken against any resource. The output of the job can then be viewed to verify the expected results.

Another method to test the results of the variable substitution is by using TYPRUN=HOLD on the JOB card (also provided in the distributed in the JCL has a comment line). This allows the variable substitution to take place and the submission of the job. The results can be viewed using the SDSF input view. You will have to issue the INPUT ON command in SDSF to view the SYSIN control statements. You will also need to have cancel authority to purge the job.

Note: TYPRUN=SCAN cannot be used. If used, SVESA will be unable to determine when the job ends.

Several of the solutions outlined here are used as part of the multi-level solutions described in "Distributed Multi-Level Automation Solutions" on page 6-26. The invocation command used in these solutions can be issued through the operator console or can be used in an AutoOPERATOR Rule fired on a timed interval. The commands themselves are examples and can be altered as necessary. Refer to Chapter 5, "Using AUTO Commands," for more information.

Delete Never Opened Solution

This solution deletes data sets on the volume that were never opened.

Function: AUTODS

Solution: DELNOPEN

Invocation Command:

F *MVSRMAS*,SVOS AUTODS VOL=*xxxxxx*,SOL=DELNOPEN

(where *MVSRMAS* is your MVSRM started task name)

Start AUTODS processing causing AUTODS records (AD_ prefixed records) to be created for volume *xxxxxx* and include DELNOPEN in their record definition during creation.

FLST

Member: SMFLSTAD

```
SET MODE=ACT
EXC AD_DSN=SYS1/
INC AD_DSN=/
```

Set to active for the inclusion of all data sets except for data sets prefixed with SYS1. Therefore, all data sets except for SYS1-prefixed data sets will be passed on for RLST processing. More than likely you will have several solutions coded in an RLST member. The way this FLST is example is coded, all SYS1-prefixed data sets will not be processed in any of the solutions coded in the RLST member for the AUTODS function. If you wanted SYS1-prefixed data sets to be excluded from processing in this solution only but included in other solutions for the AUTODS function, you would remove the EXC AD_DSN=SYS1/ statement above from the FLST and code it in the RLST as part of this solution.

RLST

Member: SMRLSTAD

```
*--------------------------------------------------------*
*    DELETE NEVER OPENED
*--------------------------------------------------------*
*
SET ACT_JOB=SRMJOB03
*SET ACT_EVENTID=AD086
EXC AD_DSORG=VS
INC SOLUTION=DELNOPEN                                 X
    AD_PUSED=0                                        X
   AD_CHG=N
*
```

This Rule list creates a SET result group (subset) of data set records passing the FLST criteria that contain DELNOPEN in their record definition that have a zero percentage used and a changed bit indicator set to N, excluding any data sets that are VSAM.

SYS1/ data sets have been filtered out from being processed in this solution by the FLST statements; however, no other high-level qualifiers have been excluded. For this solution it is possible that some data sets will pass the FLST filtering that you do not want deleted. For instance, during the installation of some subsystems dump data sets are created (like the case with CICS and DB2). These data sets remain empty (percent used zero) and unreferenced (change bit set to N) until the application using the subsystem abends, thus using the data set. Deleting this type of data set may cause a problem with your subsystem. Therefore, there may be other high-level qualifiers you should add to an EXC statement in the RLST definition (for example, if the high-level qualifier for your CICS dump data sets were “CICS,” add the exclude statement EXC AD_DSN=CICS/). Great care should be taken when implementing this solution.

JCL Member

Member: SRMJOB03

```
//SRMJOB03 JOB (XXXX),'XXXXX',CLASS=A,MSGCLASS=X
//*    TYPRUN=HOLD
//*******************************************************
//*        DELETE DATA SET *
//*******************************************************
//*        RESOURCE IS &RESTYPE.1
//*        SOLUTION IS &SOL.1
//*        FUNCTION IS &FUNC.1
//*******************************************************
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
)DO &DSN.0
//*DD&I\ DD DISP=(OLD,DELETE,DELETE),
//*         DSN=&&DSN\.&I\\,
//*         VOL=SER=&&VOLSER\.&I\\,
//*         UNIT=SYSALLDA
)END
```

The SRMJOB03 skeleton JCL member name is coded in the ACT_JOB parameter. This member needs be located in a data set within the SYSPROC DD library concatenation of your AutoOPERATOR address space. This is a basic IEFBR14 that allocates the data set and deletes it. Each record passed would cause the DD statements to be repeated as they are encompassed in the DO loop. Therefore, in this distributed solution, the end result would be the submission of one batch job that contains as many DD statements as there are data sets that passed filtering. Full explanation about skeleton tailoring is located in the IMFEXEC TAILOR command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*

DFDSS Compress and Release Solution

This solution will submit a batch job executing DFDSS compress and release against the top twenty-five non-VSAM data sets that have greater than five extents residing on the volume specified.

Invocation Command:

F *MVSRMAS*,SVOS AUTODS VOL=*xxxxxx*,SOL=COMPRLSE

where *MVSRMAS* is your MVSRM started task name

This command starts AUTODS processing, causing AUTODS records (AD_ prefixed records) to be created for volume *xxxxxx*, and includes the COMPRLSE solution in each record definition during creation.

FLST

Member: SMFLSTAD

```
SET MODE=ACT
EXC AD_DSN=SYS1/
INC AD_DSN=/
```

Set to active for the inclusion of all data sets except for data sets prefixed with SYS1. Therefore, all data sets except for SYS1-prefixed data sets will be passed on for RLST processing. More than likely you will have several solutions coded in an RLST member. The way this FLST is example is coded, all SYS1-prefixed data sets will not be processed in any of the solutions coded in the RLST member for the AUTODS function. If you wanted SYS1-prefixed data sets to be excluded from processing in this solution only but included in other solutions for the AUTODS function, you would remove the EXC AD_DSN=SYS1/ statement above from the FLST and code it in the RLST as part of this solution.

RLST

Member: SMRLSTAD

```
*-------------------------------------------------------*
*   DFDSS COMPRESS AND RELEASE SPACE
*-------------------------------------------------------*
SET SORT=(AD_EXTS,D)                      X
*   ACT_EVENTID=AD080
    ACT_JOB=SRMJOB01                      X
    ACT_COUNT=25
INC SOLUTION=COMPRLSE                     X
    AD_EXTS>2
    AD_DSORG=PO
```

This Rule list creates a SET result group (subset) of data set records passing the FLST criteria that contain COMPRLSE in the record definitions and that have extents greater than five, and that have a DSORG of PO. It sorts the SET result group by the number of extents in descending order, putting the data sets with the largest number of extents at the top of the list. It takes the first twenty-five records from the SET result group and uses them as input to the skeleton JCL.

A simple variation for this solution could be to process only ten records or to not to limit the number of records processed at all. You would do this by changing the ACT_COUNT=25 parameter to ACT_COUNT=10 or by deleting the ACT_COUNT parameter altogether.

JCL Member

Member: SRMJOB01

```
//SRMJOB01 JOB (XXXX),'XXXXX',CLASS=A,MSGCLASS=X
//*    TYPRUN=HOLD
//********************************************************
//*      DFDSS COMPRESS AND RELEASE DATA SET *
//********************************************************
//*        RESOURCE IS &RESTYPE.1
//*        SOLUTION IS &SOL.1
//*        FUNCTION IS &FUNC.1
//********************************************************
//COMPRESS EXEC PGM=ADRDSSU,REGION=0K
//SYSPRINT DD SYSOUT=*
)DO &DSN.0 I
//DD&I     DD UNIT=SYSALLDA,
//         VOL=SER=&&VOLSER\.&I\\,
//         DISP=SHR
)END
//SYSIN    DD *
)DO &DSN.0 I
```

This is the skeleton JCL member, named SRMJOB01, which is coded in the ACT_JOB parameter. This member needs be located in a data set within the SYSPROC DD library concatenation of your AutoOPERATOR address space. This is basic DFDSS JCL that issues a compress and release using the MINSEC parameter. You should review DFDSS control statements and DD statements documented in the IBM manual prior to implementing this solution.

In this example, each record passed would cause the control statements to be repeated as they are encompassed in the DO loop. Therefore, in this example, the end result would be the submission of one batch job that contained 25 DD statements matched to 25 paired compress and release statements, one for each data set. This assumes ACT_COUNT=25 is coded and there are actually 25 data sets that match the FLST/RLST criteria. For a full explanation about skeleton tailoring, see the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Migration to Level 1 Solution

This solution executes TSO commands from a batch job to migrate twenty data sets that have not been referenced in fifteen days.

Invocation Command:

```
F MVSRMAS,SVOS AUTODS VOL=xxxxxx,SOL=MIGRATLl
```

where *MVSRMAS* is your MVSRM started task name

This command starts the AUTODS processing, causing AUTODS records (AD_ prefixed records) to be created for volume *xxxxxx* and includes MIGRATL1 in each record definition during creation.

FLST

Member: SMFLSTAD

```
SET MODE=ACT
EXC AD_DSN=SYS1/
INC AD_DSN=/
```

Set to active for the inclusion of all data sets except for data sets prefixed with SYS1. Therefore, all data sets except for SYS1-prefixed data sets will be passed on for RLST processing. More than likely you will have several solutions coded in an RLST member. The way this FLST is example is coded, all SYS1-prefixed data sets will not be processed in any of the solutions coded in the RLST member for the AUTODS function. If you wanted SYS1-prefixed data sets to be excluded from processing in this solution only but included in other solutions for the AUTODS function, you would remove the EXC AD_DSN=SYS1/ statement above from the FLST and code it in the RLST as part of this solution.

RLST

Member: SMRLSTAD

```
*-------------------------------------------------------*
*    DFSMS/HSM MIGRATE
*-------------------------------------------------------*
*
SET SORT=(AD_DAYS,D)                                X
*   ACT_EVENTID=AD082                               X
    ACT_JOB=SRMJOB02                                X
    ACT_COUNT=20
EXC AD_DSORG=VS
EXC AD_DSORG=--
INC SOLUTION=MIGRATL1                               X
    AD_DAYS>15
*
```

This Rule list creates a SET result group (subset) of data set records passing the FLST criteria that contain MIGRATL1 in the record definitions and that have not been referenced in fifteen days, excluding any data sets that are VSAM and data sets where the DSORG is undetermined or the data set had not been opened. It sorts the SET result group by days not referenced in descending order, putting the data sets with the greatest number of days not referenced at the top of the list. It takes the first twenty records from the SET result group and uses them as input to the skeleton JCL.

A variation of this example could be to exclude certain data sets from being migrated. For example, to exclude data sets with the high-level qualifier of SYS2 from this solution, you would add EXC AD_DSN=SYS2/ to the RLST statements of this solution.

JCL Member

Member: SRMJOB02

```
//SRMJOB02 JOB (XXXX),'XXXXX',CLASS=A,MSGCLASS=X
//*    TYPRUN=HOLD
//*******************************************************
//*        MIGRATE ML1 *
//*******************************************************
//*        RESOURCE IS &RESTYPE.1
//*        SOLUTION IS &SOL.1
//*        FUNCTION IS &FUNC.1
//*******************************************************
//MIGRATE1 EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTERM  DD   SYSOUT=*
//SYSTSIN  DD  *
)DO &DSN.0 I
//* HMIGRATE '&&DSN\.&I\\' WAIT
)END
/*
//
```

This is the skeleton JCL member, named SRMJOB02, which is coded in the ACT_JOB parameter. This member needs be located in a data set within the SYSPROC DD library concatenation of your AutoOPERATOR address space. This executes the HMIGRATE TSO command in batch. When run against non-migrated data sets, it will migrate them to level one. The WAIT parameter keeps the job from ending until HSM has finished processing. If WAIT was not coded, the HMIGRATE commands would be issued and the job would end immediately causing job-end notification to occur before the automation against the resource actually finished processing. This would mean that subsequent automation processing against the resource could be initiated before the current (HSM) processing is finished. Processing two solutions against one resource simultaneously could alter the predetermined solution plan and alter the end results.

Each record passed would cause the control statements to be repeated as they are encompassed in the DO loop. Therefore, in this example, the end result would be the submission of one batch job that contained 20 HMIGRATE commands. This assumes ACT_COUNT=20 is coded and there are actually 20 data sets that match the FLST/RLST criteria. Full explanation about skeleton tailoring is located in the IMFEXEC TAILOR command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Delete Uncataloged Data Sets Solution

This solution will submit a batch job, executing IEFBR14, that deletes uncataloged data sets.

Invocation Command:

F *MVSRMAS*,SVOS AUTODS VOL=*xxxxxx*,SOL=DELUNCAT

where *MVSRMAS* is your MVSRM started task name

This command starts AUTODS processing, causing AUTODS records (AD_ prefixed records) to be created for volume *xxxxxx* and include DELUNCAT in each record definition during creation.

FLST

Member: SMFLSTAD

```
SET MODE=ACT
EXC AD_DSN=SYS1/
INC AD_DSN=/
```

Set to active for the inclusion of all data sets except for data sets prefixed with SYS1. Therefore, all data sets except for SYS1-prefixed data sets will be passed on for RLST processing. More than likely you will have several solutions coded in an RLST member. The way this FLST is example is coded, all SYS1-prefixed data sets will not be processed in any of the solutions coded in the RLST member for the AUTODS function. If you wanted SYS1-prefixed data sets to be excluded from processing in this solution only but included in other solutions for the AUTODS function, you would remove the EXC AD_DSN=SYS1/ statement above from the FLST and code it in the RLST as part of this solution.

RLST

Member: SMRLSTAD

```
*--------------------------------------------------------
*     DELETE UNCATLGD
*--------------------------------------------------------
*
*
SET ACT_JOB=SRMJOB03
*SET ACT_EVENTID=AD084
EXC AD_DSORG=VS
INC SOLUTION=DELUNCAT                                 X
    AD_CAT=N
*
```

This Rule list creates a SET result group (subset) of data set records passing the FLST criteria that contain DELUNCAT in the record definition and are uncataloged, excluding any data sets that are VSAM.

SYS1/ data sets have been filtered out from being processed in this solution by the FLST statements; however, no other high-level qualifiers have been excluded. It is possible that your installation shares volumes between systems that have data sets cataloged to one system but not another. As the records being processed through this solution were generated from one system only, it is possible that some data sets will pass the FLST filtering that you do not want deleted (especially those cataloged in another Master Catalog). Great care should be taken when implementing this solution.

JCL Member

Member: SRMJOB03

```
//SRMJOB03 JOB (XXXX),'XXXXX',CLASS=A,MSGCLASS=X
//*    TYPRUN=HOLD
//*******************************************************
//*        DELETE DATA SET *
//*******************************************************
//*        RESOURCE IS &RESTYPE.1
//*        SOLUTION IS &SOL.1
//*        FUNCTION IS &FUNC.1
//*******************************************************
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
)DO &DSN.0 I
//*DD&I\ DD DISP=(OLD,DELETE,DELETE),
//*         DSN=&&DSN\.&I\\,
//*         VOL=SER=&&VOLSER\.&I\\,
//*         UNIT=SYSALLDA
)END
```

This is the skeleton JCL member, named SRMJOB03, which is coded in the ACT_JOB parameter. This member needs be located in a data set within the SYSPROC DD library concatenation of your AutoOPERATOR address space. This is a basic IEFBR14 that allocated the data set and deletes it. Each record passed causes the DD statements to be repeated as they are encompassed in the DO loop. Therefore, in this example, the end result is the submission of one batch job that contains as many DD statements as there were data sets that passed filtering. Full explanation about skeleton tailoring is located in the IMFEXEC TAILOR command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs.*

DFDSS DEFRAG Solution

This solution will issue an event. The event will be captured by an AutoOPERATOR Rule. That Rule will execute a REXX EXEC that will process parameters passed to it and submit a batch job that executes DFDSS (ADRDSSU) DEFRAG against a volume.

Invocation Command:

F *MVSRMAS*,SVOS AUTOVOL POOL=*xxxxxx*,SOL=DEFRAG

where *MVSRMAS* is your MVSRM started task name

This command starts AUTOVOL processing, causing AUTOVOL records (AV_ type records) to be created for volumes in POOL *xxxxxx* and includes DEFRAG in each record definition during creation.

FLST

Member: SMFLSTAV

```
SET MODE=ACT
INC AV_VOL=/
```

Set to active for the inclusion of all volumes. Therefore all volumes will be passed on for RLST processing.

RLST

Member: SMRLSTAV

```
*--------------------------------------------------------
*
*    DEFRAG
*--------------------------------------------------------
*
*
SET ACT_EVENTID=AV092
INC SOLUTION=DEFRAG                                  X
   AV_FRAGI>100
*
```

This Rule list issues an event for each volume record passing the FLST criteria that contains DEFRAG in the record definition and have a fragmentation index greater than 100. Running a DFDSS DEFRAG against a volume or multiple volumes can, for some length of time, tie up resources critical to the operation of your installation. Great care should be taken when deciding on what resources (volumes, pools, and so on) to run this solution against and the scheduling of when this solution runs. You should review DFDSS control statements documented in the IBM manual prior to implementing this solution.

Event

Member: SMEVNT00

```
SET EVENTID=AV092
    SEV=W
    MODE=ACT
    TEXT='VOL &AV_VOL FRAG INDEX &AV_FRAGI SUB JOB     X
    SMJOB04 SOLUTION DEFRAG'
```

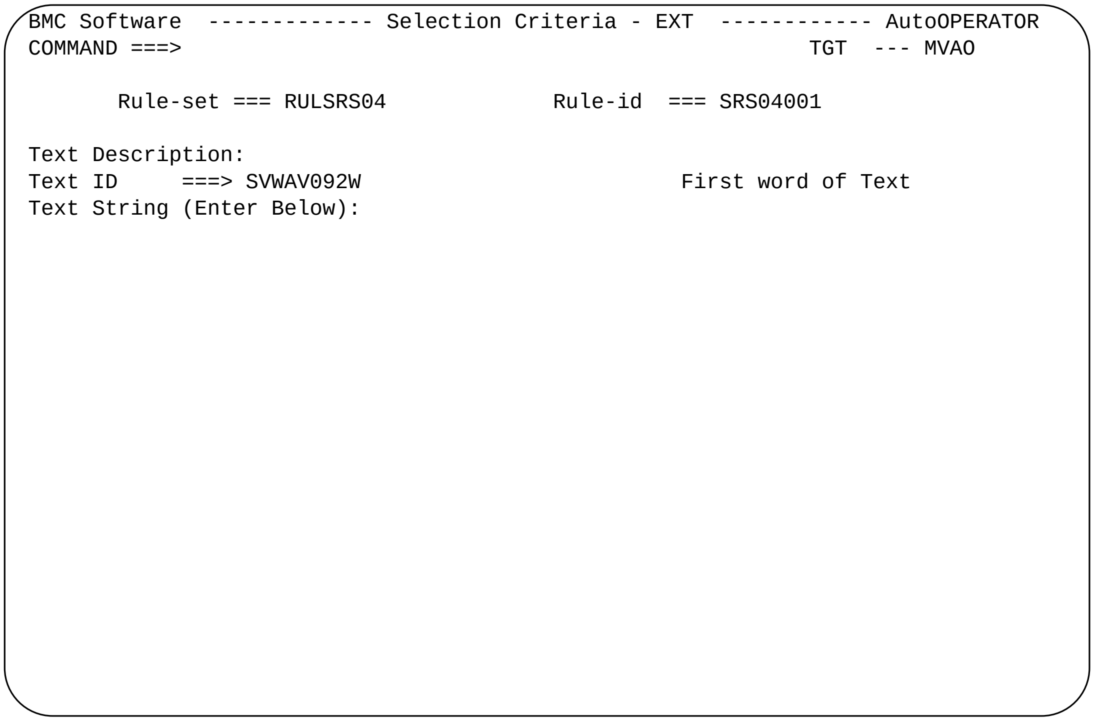
This event would be coded in your SMEVNT*xx* member. The text uses the variable replacement parameters available for the AUTOVOL record. The X is a continuation character. The resulting text for this example would be:

SVWAV092W AUTOVOL *xxxxxxxx* SJSG VOL *yyyyyy* FRAG INDEX *nnn* SUB JOB SRMJOB04 SOLUTION DEFRAG

Where, *xxxxxxxx* is the MAINVIEW SRM started task name, *yyyyyy* is the volume name and *nnn* is the fragmentation index value. Notice that *yyyyyy* is the sixth word in the event text, SRMJOB04 is the twelfth word, and DEFRAG is the fourteenth word in the event text. The position of the words is important because the words will be used in the AutoOPERATOR Rule that fires for this event. If the event text is changed, the AutoOPERATOR Rule must also be changed.

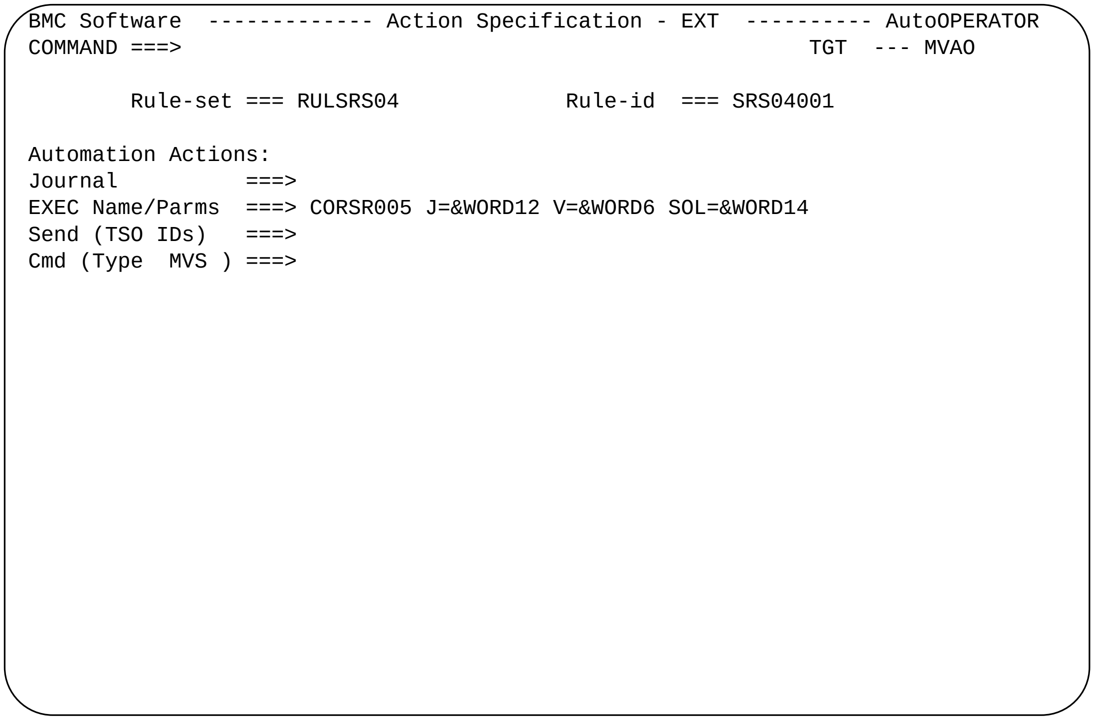
Member: RULSRS04

Figure 6-1 Selection Criteria Panel in AutoOPERATOR for this Rule



```
BMC Software  ------------- Selection Criteria - EXT  ------------ AutoOPERATOR
COMMAND ===>                                                TGT  --- MVAO

       Rule-set === RULSRS04             Rule-id  === SRS04001

Text Description:
Text ID     ===> SVWAV092W                         First word of Text
Text String (Enter Below):
```

Figure 6-2 Action Specification Panel in AutoOPERATOR for this Rule



```
BMC Software  ------------- Action Specification - EXT  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- MVAO

        Rule-set === RULSRS04             Rule-id  === SRS04001

Automation Actions:
Journal          ===>
EXEC Name/Parms  ===> CORSR005 J=&WORD12 V=&WORD6 SOL=&WORD14
Send (TSO IDs)   ===>
Cmd (Type  MVS ) ===>
```

AutoOPERATOR Rule Set RULSRS04 must be enabled. It filters on external events with the message text of SVWAV092W. When the criteria is met, the Rule fires causing REXX EXEC CORSR005 to be invoked, passing the parameters J= for the Job Name, V= for the VOLSER, and SOL= for the SOLUTION name. The variables, &WORD*nn*, are replaced by the word in the relative position number (*nn*) taken from the text of the event message, SVWAV092W. These variables are used in tailoring the skeleton JCL.

JCL Member

Member: SRMJOB04

```
//SRM&VOLSER.1:2:5\ JOB (XXXX),'XXXXX',CLASS=A,MSGCLASS=X
//*      TYPRUN=HOLD
//*******************************************************
//*        DEFRAG     *
//*******************************************************
//*        RESOURCE IS &RESTYPE.1
//*        SOLUTION IS &SOL.1
//*        FUNCTION IS &FUNC.1
//*******************************************************
//DEFRAGIT EXEC PGM=ADRDSSU,REGION=0K
//SYSPRINT DD SYSOUT=*
)DO &VOLSER.0 I
//DD&I
DDDISP=SHR,UNIT=SYSALLDA,VOL=SER=&&VOLSER\.&I\\
)END
//SYSIN    DD *
)DO &VOLSER.0 I
//* DEFRAG DDNAME(DD&I\)
)END
/*
//
```

This is the skeleton JCL member, named SRMJOB04, used in the coded Rule. This member needs be located in a data set within the SYSPROC DD library concatenation of your AutoOPERATOR address space. This is basic DFDSS JCL that issues a DEFRAG for the volumes that passed filtering. Each record passed causes the control statements to be repeated as they are encompassed in the DO loop.

You will notice that the JOB card in this example uses variable substitution in the job name. This variation of variable substitution is one of the features available in the skeleton tailoring facility. In this example the substitution will resolve the variable VOLSER and insert its contents beginning with the second character for a length of 5 characters into the output stream. Since this solution submits one job for each volume passed, each job will have a unique job name from the forth character on (SRMRES01, SRMAPVOL, and so on).

If this skeleton JCL is used together with the ACT_JOB parameter, where one job would be submitted for many volumes, the last VOLSER that passed through the tailoring process will be used in the job name.

Since this example uses ACT_EVENTID in the RLST, which will issue an event for each volume passing the criteria, the DO loop processes only once, as only one volume record will be passed to this skeleton JCL for processing at a time. This causes the creation and submission of one batch job for each event (volume). Each time the SVWAV902W is issued, Rule SRS04001 fires. Each time the Rule SRS04001 fires, the REXX EXEC CORSR005 is invoked, accepting one set of variables taken from the event text. A one-to-one relationship flows through the process. If ACT_JOB=SRMJOB04 was coded in the RLST instead of ACT_EVENTID, a SET result group would be created during processing, passing a group of volume records at one time to be used in the skeleton tailoring process. A many-to-one relationship flows through this process. The end result would be the submission of one batch job that contains as many control statements initiating a DEFRAG against as many volumes that passed filtering.

There are a couple noteworthy items in this example. One is that, although only one record is used in each batch job that is submitted, the skeleton JCL is coded using DO loops. Coding skeleton JCL using DO loops gives you the flexibility to use the same skeleton JCL for ACT_JOB processing or ACT_EVENTID processing (assuming the event causes CORSR005 to be invoked). Full explanation about skeleton tailoring is located in the IMFEXEC TAILOR command in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Another note is that even though this solution submits a batch job, the fact that it does so using the CORSR005 REXX EXEC means that the MAINVIEW SRM address space will not be notified when that job completes. Therefore, if the invocation of CORSR005 is used as part of an MLA solution, it must be the last action the MLA solution takes.

StorageGUARD Net Capacity Load Solution

This solution will issue an event. The event will be captured by an AutoOPERATOR Rule. That Rule will issue an AutoOPERATOR Alert. This solution differs from the others in that it does not initiate the generation of records through a modify command. The records used in this solution are generated through StorageGUARD Historical Performance. At the interval when StorageGUARD creates and logs these records, they will be passed through the FLST/RLST processing. This solution requires that you are licensed for StorageGUARD, your StorageGUARD Historical Performance is running, and that function SGPRSFRC be listed in your SMFUNC*xx* member and is coded as active.

FLST

Member: SMFLST34

```
SET MODE=ACT
INC SGP_RSFNAME=*
```

Set to active for the inclusion all IXPF subsystem frame names. Therefore, all IXPF subsystem frame names will be passed on for RLST processing.

RLST

Member: SMRLST34

```
*-----------------------------------------------------------
*     SEND EVENTS FOR NET CAPACITY LOAD GREATER THAN 85%
*-----------------------------------------------------------
*
SET EVENTID=SG092
INC SGP_NCLPERC > 85
```

This Rule list issues an event for each SGPRSFRC record passing the FLST criteria that contains a value greater than 85 for the SGP_NCLPERC (Net capacity load percent) field. Since the FLST is not filtering out any IXFP frame names, every SGPRSFRC will be passed to this RLST for processing.

Event

Member: SMEVNT00

```
-------------------------------------------------------------------------
SET EVENTID=SG092
    SEV=W
    MODE=ACT
    TEXT='NET CAPACITY LOAD PERCENT IS &SGP_NCLPERC FOR FRAME           X
&SGP_RSFNAME'
-------------------------------------------------------------------------
```

This event would be coded in your SMEVNTxx member. The text uses the variable replacement parameters available for the SGPRSFRC record. The X is a continuation character. The resulting text for this example would be:

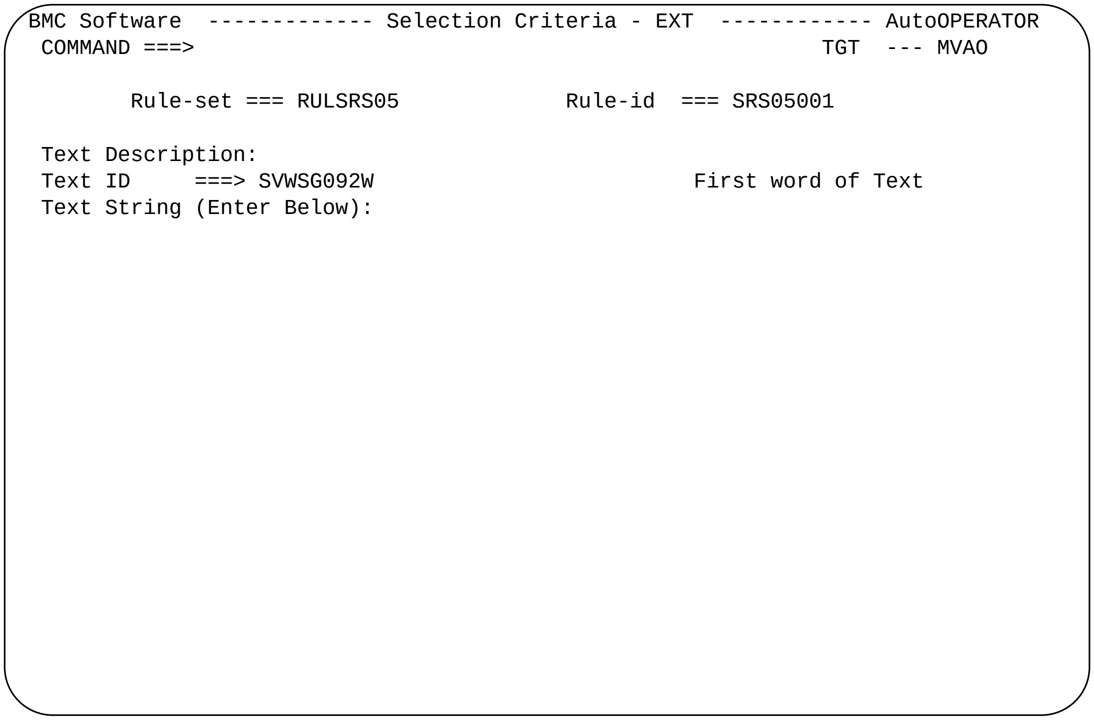
```
SVWSG092W SGPRSFRC aaaaaaa bbbb NET CAPACITY LOAD PERCENT
is ccc FOR FRAME dddddddd
```

Where, *aaaaaaaa* is the MAINVIEW SRM started task name, *bbbb* is the system ID, ccc is the NCL percent and *dddddddd* is the frame name. Notice that *ccc* is the tenth word in the event text and *dddddddd* is the thirteenth. The position of the words is important because the words will be used in the AutoOPERATOR Rule that fires for this event. If the event text is changed, the AutoOPERATOR Rule must also be changed.

Rule Set

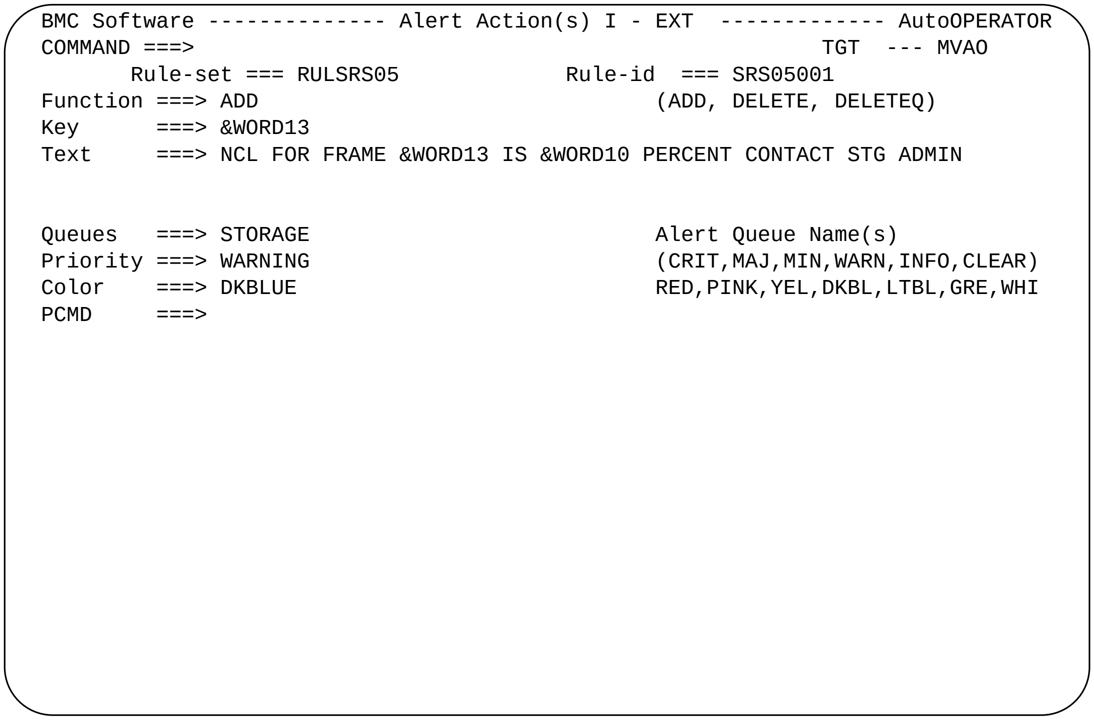
Member: RULSRS05

Figure 6-3 Selection Criteria Panel in AutoOPERATOR for this Rule



```
BMC Software  ------------- Selection Criteria - EXT  ------------ AutoOPERATOR
 COMMAND ===>                                                 TGT  --- MVAO

        Rule-set === RULSRS05             Rule-id  === SRS05001

 Text Description:
 Text ID     ===> SVWSG092W                          First word of Text
 Text String (Enter Below):
```

Figure 6-4 Alert Action Panel in AutoOPERATOR for this Rule



```
BMC Software -------------- Alert Action(s) I - EXT  ------------- AutoOPERATOR
COMMAND ===>                                                TGT  --- MVAO
       Rule-set === RULSRS05             Rule-id  === SRS05001
Function ===> ADD                              (ADD, DELETE, DELETEQ)
Key      ===> &WORD13
Text     ===> NCL FOR FRAME &WORD13 IS &WORD10 PERCENT CONTACT STG ADMIN


Queues   ===> STORAGE                          Alert Queue Name(s)
Priority ===> WARNING                          (CRIT,MAJ,MIN,WARN,INFO,CLEAR)
Color    ===> DKBLUE                           RED,PINK,YEL,DKBL,LTBL,GRE,WHI
PCMD     ===>
```

Distributed Multi-Level Automation Solutions

The solutions in the previous section initiate a function and, based on the results, take a single action (a solution) for that function. The examples in this section take this process a step further. By using the AUTOLEV parameter and additional Rules, the initiated function can incorporate several solutions working in succession to achieve a desired result.

Pool Space Management Automation Solution

Function: AUTOPOOL

Solution: SRS03002

Description:

This distributed solution automates pool space management. In this distributed solution, pools beginning with WRK* are examined on a fifteen minute interval. For any WRK pool that is over 90% full, actions are taken until the pool is below 85% full. These actions are taken in a tiered approach using multi-level automation. The distributed solution includes use of AUTODS solutions.

AutoOPERATOR Rules involved in this solution are contained in Rule Set RULSRS03. This distributed solution demonstrates job submission and MLA usage on a pool resource. As job submission is used, the Rules in Rule Set RULSRS01 that detect job end, must also be enabled.

Automation Flow:

In the distributed solution, the solution is initiated on a time based AutoOPERATOR Rule. The Rule fires every 15 minutes, between 8 A.M. and 8 P.M., and requests the SRS03002 AUTOPOOL solution for all WRK* pools. MLA=Y is specified to indicate multi-level automation.

For any pool over 90% full, event AP092 is generated. AutoOPERATOR Rules then fire on the AP092 event, kicking off the following solutions in the order listed:

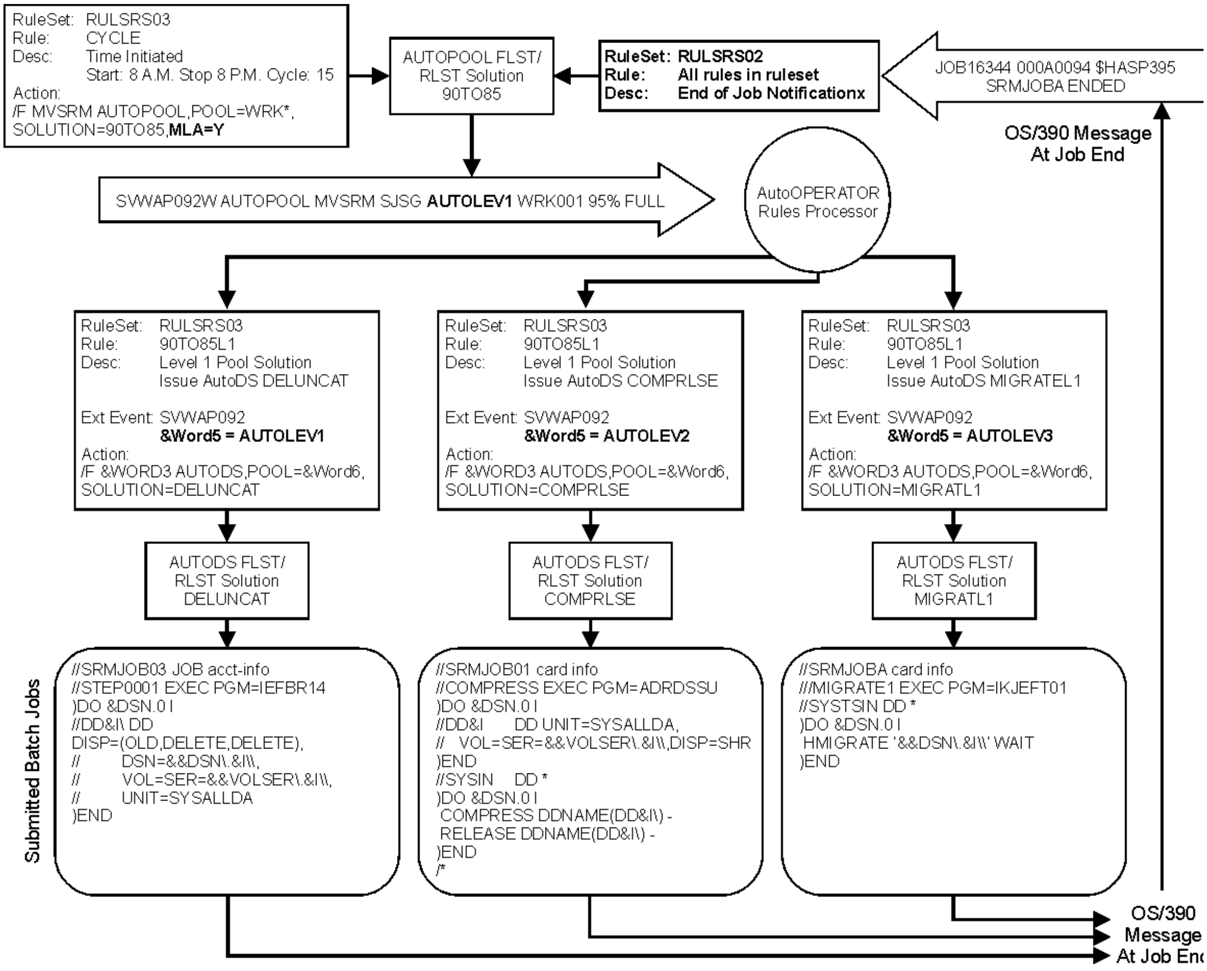
1. AUTODS Solution DELUNCAT. This deletes uncataloged data sets in the pool.

2. AUTODS Solution COMPRLSE. This compresses selected data sets and releases free space.

3. AUTODS Solution MIGRATL1. This migrates selected data set to level one.

The AP092 event fires at the completion of each solution until the pool is below 85% full. Once the pool falls below 85% used, or all three levels have been initiated, automation stops.

Figure 6-5 Pool Space Management Automation Solution Diagram



```
RuleSet:  RULSRS03
Rule:     CYCLE
Desc:     Time Initiated
          Start: 8 A.M. Stop 8 P.M. Cycle: 15
Action:
/F MVSRM AUTOPOOL,POOL=WRK*,
SOLUTION=90TO85,MLA=Y
```

```
RuleSet:  RULSRS02
Rule:     All rules in ruleset
Desc:     End of Job Notificationx
```

```
JOB16344 000A0094 $HASP395
SRMJOBA ENDED
```

```
SVWAP092W AUTOPOOL MVSRM SJSG AUTOLEV1 WRK001 95% FULL
```

```
RuleSet:  RULSRS03
Rule:     90TO85L1
Desc:     Level 1 Pool Solution
          Issue AutoDS DELUNCAT

Ext Event SVWAP092
          &Word5 = AUTOLEV1
Action:
/F &WORD3 AUTODS,POOL=&Word6,
SOLUTION=DELUNCAT
```

```
RuleSet:  RULSRS03
Rule:     90TO85L1
Desc:     Level 1 Pool Solution
          Issue AutoDS COMPRLSE

Ext Event SVWAP092
          &Word5 = AUTOLEV2
Action:
/F &WORD3 AUTODS,POOL=&Word6,
SOLUTION=COMPRLSE
```

```
RuleSet:  RULSRS03
Rule:     90TO85L1
Desc:     Level 1 Pool Solution
          Issue AutoDS MIGRATEL1

Ext Event SVWAP092
          &Word5 = AUTOLEV3
Action:
/F &WORD3 AUTODS,POOL=&Word6,
SOLUTION=MIGRATL1
```

```
//SRMJOB03 JOB acct-info
//STEP0001 EXEC PGM=IEFBR14
)DO &DSN.0 I
//DD&I\ DD
DISP=(OLD,DELETE,DELETE),
//        DSN=&&DSN\.&I\\,
//        VOL=SER=&&VOLSER\.&I\\,
//        UNIT=SYSALLDA
)END
```

```
//SRMJOB01 card info
//COMPRESS EXEC PGM=ADRDSSU
)DO &DSN.0 I
//DD&I       DD UNIT=SYSALLDA,
//   VOL=SER=&&VOLSER\.&I\\,DISP=SHR
)END
//SYSIN    DD *
)DO &DSN.0 I
 COMPRESS DDNAME(DD&I\) -
 RELEASE DDNAME(DD&I\) -
)END
/*
```

```
//SRMJOBA card info
///MIGRATE1 EXEC PGM=IKJEFT01
//SYSTSIN DD *
)DO &DSN.0 I
 HMIGRATE '&&DSN\.&I\\' WAIT
)END
```

Elements:

AutoOPERATOR Rules: SRS02001 (optional Rule)

Used mostly for testing solutions and automation flows, its purpose is to write any SVW* external event received (that is, SVESA event) to the AutoOPERATOR journal. The AutoOPERATOR journal can then be used to examine the exact events that were sent by SVESA in an automation process.

This Rule is contained in Rule Set RULSRS02. Note this Rule Set is filtered so that the Rules contained in it are only examined for external event types, and then only when the first word of the event is SVW*. This filtering can be used on all Rule Sets containing Rules that act upon SVESA events, as SVESA events are always type of external and begin with SVW*. This usage of Rule Set filters is more efficient and can provide better performance than the same filters within each Rule of the Rule Set.

AutoOPERATOR Rule: SRS03001

This Rule initiates the automation of the WRK pools. This is a time driven Rule, scheduled to fire every fifteen minutes between the hours of 8 A.M. and 8 P.M. The AUTOPOOL command specifies POOL=WRK*, requesting SOLUTION=90TO85. Any pool over 90% full will be reduced to 85% full.

AUTOPOOL FLST

This solution does not contain AUTOPOOL FLST statements. The AUTOPOOL FLST member can be used to exclude a pool or pools from all AUTOPOOL solutions. The distributed FLST member includes all pools. If any WRK* pools are excluded in FLST they will not be processed by this, or any other, AUTOPOOL solution.

AUTOPOOL RLST Solution: 90TO85

This RLST solution encompasses two set statements. These two RLST SET statements are used in MLA processing, the first handles the selection of pools to start the MLA solution sequence on, and the second handles selection of pools to stop the sequence on. Pools are selected to begin this process when they are over 90% full. The AUTOPOOL console command specifies MLA=Y, and therefore, as each action taken on the pool completes, this AUTOPOOL SET solution is checked again, as part of the MLA process to determine whether to take another action on the pool. The second SET statement causes actions to be taken on the pool until it is below 85% utilization.

The end result of requesting this solution is that event AP092 is issued for a volume once it is over 90% full, and continues to do so with each automation level until the volume is less than 85% full.

This selection/filtering is done by use of the AUTOLEV filter and requires a understanding of FLST/RLST statement processing. This AUTOLEV field is fully documented in “Multi-Level Automation (MLA) Processing” on page 4-35. The field indicates the current automation level being performed. The first SET statement filters on AUTOLEV=AUTOLEV1 and pool %full greater than 90. The action, specified by ACT_EVENTID=AP092, causes event AP092 to be generated when the pool is over 90% full on the first automation level. On subsequent automation levels this first SET statement will not match the pool resource, as the pool will have an AUTOLEV value of AUTOLEV2 or higher. In this situation the next SET statement will be inspected, as dictated by FLST/RLST processing. This second SET statement filters on AUTOLEV > AUTOLEV1 and pool %full greater than 85%. The action of generating event AP092 will occur on any subsequent automation level if the pool is still over 85% full.

If this solution is only invoked from MLA type requests, then the filter of the AUTOLEV>AUTOLEV1 in the second SET statement is superfluous, as the statement will only be seen if the AUTOLEV value of the resource is not AUTOLEV1. However, if this solution is invoked without MLA processing then the AUTOLEVL value will be AUTOLEV0, in which case this filter is not superfluous as it prevents non-MLA requests from matching this SET. This solution requires MLA usage, and therefore this filter is specified to enforce that requirement.

SMEVNT00 Event: AP092

This event definition creates the SVWAP092 event message text and activates the AP092 event. The automation level, pool name, and percentage full, and volume count fields are specified as variables in the text, to be substituted into the event text whenever the event is generated by the AUTOPOOL function.

The automation level and pool name fields are used in the AutoOPERATOR Rule SRS03002. That Rule uses the fifth word and 6th words of the event message text, and expects them to be the automation level and the pool name. Event text after these two fields may be modified, however, do not change the first two words of the text.

AutoOPERATOR Rule: SRS03002

This AutoOPERATOR Rule initiates the first action in this multi-level automation process. The Rule fires when the SVWAP092 event message is received, indicating a pool is exceeding a threshold, and the automation level is one (AUTOLEV1) indicating the first level of automation. The SVWAP092 event message is expected to contain the automation level as the 5th word and the pool name in the 6th word. No other fields in the event text are used.

Action: This Rule issues the AUTODS command for the pool, requesting the DELUNCAT solution. This solution is defined elsewhere, however, causes a job to be submitted to delete uncataloged data sets.

When the job submitted by DELUNCAT solution ends, since MLA=Y was specified on the initial AUTOPOOL request for the resource, the automation level will be incremented and the initial request for AUTOPOOL 90TO85 solution redriven for the pool. The AUTOPOOL solution will again cause event AP092 if the pool is still over 85% full. However, this time the automation level field, in FLST/RLST and the AP092 event message text, will be AUTOLEV2.

AutoOPERATOR Rule: SRS03003

This AutoOPERATOR Rule initiates the 2nd action in this multi-level automation process. The Rule fires when the SVWAP092 event message is received, indicating a pool is exceeding a threshold, and the automation level is two (AUTOLEV2) indicating the 2nd level of automation. The SVWAP092 event message is expected to contain the automation level as the 5th word and the pool name in the 6th word. No other fields in the event text are used.

Action: This Rule issue the AUTODS command for the pool, requesting the COMPRLSE solution. This solution is defined elsewhere, however, causes a job to be submitted to compress and release free space in data sets.

When the job submitted by COMPRLSE solution ends, since MLA=Y was specified on the initial request for the resource, the automation level will be incremented and the initial request for AUTOPOOL 90TO85 solution is redriven for the pool. The AUTOPOOL solution will again cause event AP092 if the pool is still over 85% full. However, this time, the automation level field, in FLST/RLST and the AP092 event message text, will be AUTOLEV3.

AutoOPERATOR Rule: SRS03004

This AutoOPERATOR Rule initiates the 3rd action in this multi-level automation process. The Rule fires when the SVWAP092 event message is received, indicating a pool is exceeding a threshold, and the automation level is three (AUTOLEV3) indicating the third level of automation. The SVWAP092 event message is expected to contain the automation level as the 5th word and the pool name in the 6th word. No other fields in the event text are used.

Action: This Rule issue the AUTODS command for the pool, requesting the MIGRATEL1 solution. This solution is defined elsewhere, however, causes a job to be submitted to migrate selected data sets in the pool, by days-unreferenced.

When the job submitted by the MIGRATL1 solution ends, since MLA=Y was specified on the initial request for the resource, the automation level will be incremented and the initial request for AUTOPOOL 90TO85 solution is redriven for the pool. The AUTOPOOL solution will again cause event AP092 if the pool is still over 85% full. However, this time, the automation level field, in FLST/RLST and the AP092 event message text, will be AUTOLEV4.

As there are no AutoOPERATOR Rules that fire with AUTOLEV4, automation ends.

Requirements:

All solutions invoked here are distributed solutions and must be installed in the appropriate AUTOPOOL or AUTODS function FLST/RLST members.

As this distributed solution uses multi-level automation, the Rules distributed in RULSRS01 must be enabled to cause SVOS notification of a submitted automation job ending. See “Job Submission and Skeleton Tailoring” on page 4-29 for requirements on job names and the usage instructions for these Rules.

AutoOPERATOR must be version 6.2 or higher as required for job submission capabilities.

Customization Tips:

Other solutions maybe added or substituted anywhere in the chain of actions.

Change the STC name in Rule SRS03001 to match the STC name of your SVOS.

Installation:

Rule Sets:	RULSRS01 for job end notification RULSRS03 for Rules associated with Pool MLA Sample
FLST/RLST:	AUTOPOOL 90TO85 Solution AUTODS DELUNCAT Solution AUTODS COMPRLSE Solution AUTODS MIGRATL1 Solution
Events:	AP092
Skeleton JCL:	See the AUTODS Solutions in the previous automation solution section.

Volume Space Management Automation Solution

This distributed solution automates volume space management.

Function: AUTOVOL

Solution: VOLGT85

Description:

In this distributed solution, volumes beginning with WRK* are examined on a 4-hour interval. For any WRK volume that is over 85% full, actions are taken until the volume is below 85% full. These actions are taken in a tiered approach using multi-level automation. This solution includes use of AUTODS solutions.

AutoOPERATOR Rules involved in this solution are contained in Rule Set RULSRS06. This distributed solution demonstrates job submission and MLA usage on a volume resource. As job submission is used, the Rules in Rule Set RULSRS01, which detect job end, must also be enabled.

Automation Flow:

In the distributed solution, the solution is initiated on a time based AutoOPERATOR Rule. The Rule fires every 4 hours, between 8 A.M. and 8 P.M. and requests the VOLGT85 AUTOVOL solution for all WRK* volumes. MLA=Y is specified to indicate multi-level automation.

For any volume over 85% full, event AV094 is generated. AutoOPERATOR Rule then fires on the AV094 event, kicking off the following solutions in the order listed:

1. AUTODS solution DELNOPEN deletes data sets on the volume that were never opened.

2. AUTODS solution DELUNCAT deletes data sets on the volume that are not cataloged.

3. The last solution is to issue an Alert within AutoOPERATOR.

Elements:

AutoOPERATOR Rules: SRS02001

This optional Rule is used mostly for testing solutions and automation flows. Its purpose is to write any SVW* external event received, that is, an Enterprise Storage Automation event, to the AutoOPERATOR journal. The AutoOPERATOR journal can then be used to examine the exact events that were sent by Enterprise Storage Automation in an automation process.

This Rule is contained in Rule Set RULSRS02. Note this Rule Set is filtered so that the Rules contained in it are only examined for external event types and then only when the first word of the event is SVW*. This filtering can be used on all Rule Sets containing Rules that act on Enterprise Storage Automation events, because these events are always external and begin with SVW*. This usage of Rule Set filters is more efficient and can provide better performance than the same filters within each Rule of the Rule Set.

AUTOVOL FLST

This solution does not contain AUTOVOL FLST statements. The AUTOVOL FLST member can be used to exclude a volume or volumes from all AUTOVOL solutions. The distributed FLST member includes all volumes. If any WRK* volumes are excluded in FLST they will not be processed by this, or any other, AUTOVOL solution.

AutoOPERATOR Rule: SRS06001

Member: RULSRS06

This Rule initiates the automation of the WRK volumes. This is a time driven Rule, scheduled to fire every hour between the hours of 8 A.M. and 8 P.M. The AUTOVOL command specifies VOL=WRK*, requesting SOLUTION=VOLGT85 and MLA=Y indicating it is processing as a multi-level solution.

AUTOVOL RLST Solution: VOLGT85

Member: SMRLSTAV

The AV094 event will be issued for every volume record where WRK is the volume prefix and the space utilization is greater than 85 percent. The volume record must also contain an AUTOLEV value greater than AUTOLEV0. The filtering criteria used here resembles most of the basic solutions except for the AUTOLEV value. This solution is matching on AUTOLEV greater than AUTOLEV0. For AUTOLEV to be greater than AUTOLEV0 it must be part of a multi-level request. Automation commands that are invoked without MLA=Y start with the AUTOLEV value set to AUTOLEV0. Automation commands invoked with MLA=Y start with the AUTOLEV value AUTOLEV1.

SMEVNT00 Event: AV094

Member: RULSRS06

This event definition creates the SVWAV094 event message text and activates the AV094 event. The automation level, volume name, and percentage full are specified as variables in the text to be substituted into the event text whenever the event is generated.

The automation level and volume name fields are used in the AutoOPERATOR Rules SRS06002, SRS06003, and SRS06004. These Rules use the third, fifth, seventh, and ninth words of the event message text. It expects these words to be the started task name, automation level, volume name, and percent full. Event text can be altered: however, the Rule firing on this event expects the fields within the text to be in a specific order. The first four words in the event text are static and cannot be changed. This includes the MAINVIEW SRM Started Task name which is the third word of the event text. Text following the TEXT= parameter in the event definition start as the fifth word.

AutoOPERATOR Rule: SRS06002

This AutoOPERATOR Rule initiates the first action in this multi-level automation process. The Rule fires when the SVWAV094W event message is received, indicating a volume is exceeding a threshold, and the automation level is one (AUTOLEV1), which indicates the first level of automation. The SVWAV094W event message is expected to contain the started task name as the third word in the text, the automation level in the fifth word, and the volume name in the seventh word. No other fields in the event text are used.

Action: This Rule issues the AUTODS command for the volume, requesting the DELNOPEN solution. This solution, defined on page 6-5, causes a job to be submitted to delete data sets that have not been opened.

When the job submitted by DELNOPEN solution ends, since MLA=Y was specified on the initial AUTOVOL request for the resource, the automation level will be incremented and the initial request for AUTOVOL VOLGT85 solution redriven for the volume. The AUTOVOL solution will again cause event AV094 if the pool is still over 85% full. However, this time the automation level field in the AV094 event message text will be AUTOLEV2.

AutoOPERATOR Rule: SRS06003

Member: RULSRS06

This AutoOPERATOR Rule initiates the second action in this multi-level automation process. The Rule fires when the SVWAV094W event message is received, indicating a volume is exceeding a threshold, and the automation level is two (AUTOLEV2) indicating the second level of automation. The SVWAV094W event message is expected to contain the started task name as the third word in the text, the automation level as the fifth word and the volume name in the seventh word. No other fields in the event text are used.

Action: This Rule issues the AUTODS command for the volume, requesting the DELUNCAT solution. This solution, defined on page 6-13, causes a job to be submitted to delete uncataloged data sets.

When the job submitted by DELUNCAT solution ends, since MLA=Y was specified on the initial request for the resource, the automation level will be incremented and the initial request for AUTOVOL VOLGT85 solution is redriven for the volume. The AUTOVOL solution will again cause event AV094 if the volume is still over 85% full. However, this time, the automation level field, in the AV094 event message text will be AUTOLEV3.

AutoOPERATOR Rule: SRS06004

Member: RULSRS06

This AutoOPERATOR Rule initiates the third action in this multi-level automation process. The Rule fires when the SVWAV094 event message is received, indicating a volume is exceeding a threshold, and the automation level is three (AUTOLEV3), which indicates the third level of automation. The SVWAV094W event message is expected to contain the started task name in the third word in the text, the automation level in the fifth word, the volume name in the seventh word, and the percent full in the ninth word. No other fields in the event text are used.

Action: This Rule issues an Alert to AutoOPERATOR. The Alert is setup to use the seventh and ninth word from the AV094 event text, that is, volume name and percent full, respectively. For the KEY field in the Alert, the volume name is appended to VOL. This makes the KEY field unique, allowing each volume to be listed in the Alert display. The percent full is used as text within the Alert. Alerts are explained in the *MAINVIEW AutoOPERATOR Basic Automation Guide*.

Since this action issues an Alert, the chain of MLA processing has ended for this volume.

Requirements:

- All solutions invoked here are distributed solutions and must be installed in the appropriate AUTOVOL and AUTODS function FLST/RLST members.
- Because this distributed solution uses job submission, the Rules distributed in RULSRS01 must be enabled to cause SVOS notification of a submitted automation job ending. See the “Job Submission and Skeleton Tailoring” on page 4-29 for requirements on job names and the usage instructions for these Rules.
- AutoOPERATOR must be version 6.2 or higher as required for job submission capabilities.

Customization Tips:

Other solutions may be added or substituted anywhere in the chain of actions.

Change the STC name in Rule SRS06001 to match the STC name of your SVOS.

Installation:

Rule Sets:	RULSRS01 for job end notification RULSRS06 for Rules associated with Vol MLA Sample
FLST/RLST:	AUTOVOL VOLGT85 Solution AUTODS DELNOPEN Solution AUTODS DELUNCAT Solution
Events:	AV094
Skeleton JCL:	See the AUTODS Solutions in the previous automated solution section.

Distributed HSM Solutions

HSM automation events are generated by AutoOPERATOR Rules. A Rule is a two-part conditional statement. When the conditions specified in the first part of the statement are met, the actions in the second part of the statement are performed. Some Rules use variables and some are time-initiated. Rules are organized into sets of related Rules in the AutoOPERATOR Rules Processor. RULSRS is the AutoOPERATOR Rule Set for predefined Enterprise Storage Automation solutions.

The predefined storage events are listed in the following table. Rules can be modified through the AutoOPERATOR Rules Processor. For more information, see the *MAINVIEW AutoOPERATOR Basic Automation Guide.*

Table 6-2 Predefined Storage Events

Events	AutoOPERATOR Parmlib Member
Initialization Solution	SRSVARG
HSM Held Resource Solution	SRSVARH
HSM Waiting Solution	SRSVARW
HSM Duplicate Request Solution	SRSVARD
HSM Commands and Replies Solution	none
HSM Alert Solution	none
HSM Message Suppression Solution	none
ADSM Hung Session Solution	SRSVARD

Note: AutoOPERATOR parmlib member SRSVARG for the Initialization solution is a prerequisite for all of the other solutions.

If you modify parameters in an AutoOPERATOR parmlib member, you can reset variable values by using the following AutoOPERATOR console command:

```
F AOAS,%SRSVAR xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
```

The variable *xxxxxxxx* represents the name(s) of the parmlib member(s) that contains the changed values.

Initialization Solution

The Initialization solution is used to initialize variables for the following solutions described in this chapter:

- HSM Held Resource
- HSM Waiting
- HSM Duplicate Request
- ADSM Hung Session

Prerequisites

There are no prerequisites for this solution.

Invocation

The Initialization solution is invoked when MVS issues the following message:

```
PM0010I BBI INITIALIZATION COMPLETE
```

The Rule that invokes this solution runs an EXEC (SRSVAR) that reads parmlib members containing solution variables and sets the variables.

Rules

This solution uses the following Rule:

SRSVAR traps BBI startup message PM0010I and calls the SRSVAR EXEC

This Rule is distributed as enabled in Rule Set RULRSRS.

EXECs

This solution uses the following EXEC:

SRSVAR reads parmlib members and sets variables; any parameters that are passed are appended with the characters VAR to complete the name of the member to be read

Parameters

The parmlib member for this solution is SRSVARG. The parameters in SRSVARG apply to multiple solutions.

Parameter	Description
HSM_Jobname	job name of the HSM address space
SV_AlertQ	queue name for all MAINVIEW SRM alerts
FOCALPT	AutoOPERATOR target name to which all alerts are routed; if you specify an * (asterisk) on this parameter, all alerts will be created on the local system

HSM Held Resource Solution

The HSM Held Resource solution issues the HSM QUERY ACTIVE command to check for held resources and for the largest free areas above and below the line. When an automated action is held, an event will be generated and displayed in the AutoOPERATOR Active Alerts display.

Prerequisites

Initialization solution

Invocation

This solution is invoked by a time-initiated Rule. It can also be invoked manually by issuing the following AutoOPERATOR console command:

%HSMHELD

Rules

This solution uses the following Rule:

HSMHELD time-initiated Rule that executes every hour at one minute past the hour

If you want to change the interval, use the AutoOPERATOR Rules application to change it.

This Rule is distributed as enabled in Rule Set RULRSRS.

EXECs

This solution uses the following EXECs:

HSMHELD issues the HSM QUERY ACTIVE command and parses through the output looking for any resource that is in a HELD status

For each resource found in a HELD status, HSMHELD issues an event of the priority that you specify. It also compares the largest free areas above and below the line to the parameters you specify and issues an event for each if the area is smaller than the threshold.

HSMRLSE called when a you respond to an event

If you respond to an event with an R, the held resource will be released and the event will be deleted. If you respond with anything else, an event that states the reply was invalid will be issued.

Parameters

The parmlib member for this solution is SRSVARH. The parameters are read by the Initialization solution and saved as shared variables.

Parameter	Description
HSMHELD_Debug	sets verbose mode for the solution 0 = off (normal) 0 = off (normal) 2 = very verbose
HSMHELD_Alert_ Priority	priority for all alerts created by this solution for a held resource or insufficient free space; values are CRITICAL, MAJOR, MINOR, WARN, INFO, or CLEARING.
HSM_Free_Below	threshold size in kilobytes below the line If the largest free area below the line is smaller than this value, this solution will issue an alert. If the value is zero (0), no checking will be performed.
HSM_Free_Above	threshold size in kilobytes above the line If the largest free area above the line is smaller than this value, this solution will issue an alert. If the value is zero (0), no checking will be performed.

HSM Waiting Solution

The HSM Waiting solution issues the HSM QUERY WAITING command to check the number of waits on each resource against the parameters you specify. You can establish warning thresholds for the following conditions:

- MIGRATE waits
- RECALL waits
- DELETE waits
- BACKUP waits
- RECOVER waits
- COMMAND waits
- ABACKUP waits
- ARECOVER waits
- TOTAL waits

If any of the values you specify are exceeded, a warning condition is issued. If you specify a zero (0) value for any condition, a threshold warning will not be issued.

Prerequisites

Initialization solution

Invocation

This solution is invoked by a time-initiated Rule. It can also be invoked manually by issuing the following AutoOPERATOR console command:

%HSMWAIT

Rules

This solution uses the following Rule:

HSMWAIT time-initiated Rule that executes every hour at one minute past the hour

If you want to change the interval, use the AutoOPERATOR Rules application to change it.

This Rule is distributed as enabled in Rule Set RULRSRS.

EXECs

This solution uses the following EXEC:

HSMWAIT issues the HSM QUERY WAITING command and parses through the output comparing the number of waits for each resource against the threshold you specify for the resource

Each time HSMWAIT finds a value over the threshold, it issues an alert.

Parameters

The parameters for this solution are in parmlib member SRSVARW. The parameters are read by the Initialization solution and saved as shared variables.

Parameter	Description
HSMWAIT_Debug	sets verbose mode for the solution 0 = off (normal) 1 = verbose 2 = very verbose
HSMWAIT_Alert_Priority	priority for all alerts created by this solution when a waiting value is over the threshold; values are CRITICAL, MAJOR, MINOR, WARN, INFO, or CLEARING.
HSMWAIT_MIGRATE	threshold for MIGRATE waits
HSMWAIT_RECALL	threshold for RECALL waits
HSMWAIT_DELETE	threshold for DELETE waits
HSMWAIT_BACKUP	threshold for BACKUP waits
HSMWAIT_RECOVER	threshold for RECOVER waits
HSMWAIT_COMMAND	threshold for COMMAND waits
HSMWAIT_ABACKUP	threshold for ABACKUP waits
HSMWAIT_ARECOVER	threshold for ARECOVER waits
HSMWAIT_TOTAL	threshold for TOTAL waits

HSM Duplicate Request Solution

The HSM Duplicate Request solution issues an HSM QUERY REQUEST command to check for duplicate requests. When a duplicate request is found, it is either canceled or an alert is issued, depending on what you specify.

Prerequisites

Initialization solution

Invocation

This solution is invoked by a time-initiated Rule. It can also be invoked manually by issuing the following AutoOPERATOR console command:

```
%HSMDUPR
```

Rules

This solution uses the following Rule:

HSMDUPR time-initiated Rule that executes every hour at one minute past the hour

If you want to change the interval, use the AutoOPERATOR Rules application to change it.

This Rule is distributed enabled in Rule Set RULRSRS.

EXECs

This solution uses the following EXEC:

HSMDUPR issues the HSM QUERY REQUEST command and parses through the output comparing requests

Each time HSMDUPR finds more than one request for the same data set, an alert is issued (if in test mode) or the second request is cancelled.

Parameters

The parameters for this solution are in parmlib member SRSVARD. The parameters are read by the Initialization solution and saved as shared variables.

Parameter	Description
HSMDUPR_Debug	sets verbose mode for the solution 0 = off (normal) 1 = verbose 2 = very verbose
HSMDUPR_Alert_Priority	priority for all alerts created by this solution for duplicate requests; values are: CRITICAL, MAJOR, MINOR, WARN, INFO, or CLEARING.
HSMDUPR_Test	sets test mode. 0 = off 1 = on When off, this solution will issue an alert instead of canceling the duplicate request.

HSM Commands and Replies Solution

The HSM Commands and Replies solution traps write-to-operator-with-reply (WTOR) messages issued by HSM and automates replies in response to them.

Prerequisites

Initialization solution

Invocation

This solution is invoked when HSM issues the WTOR messages to be trapped.

Rules

This solution uses the following Rules:

ARC0310A replies Y to the following WTOR message: ARC0310A CAN TAPE BE FOUND AND MOUNTED? REPLY Y OR N

ARC0314A replies Y to the following WTOR message: ARC0314A CAN THE n VOLUME(S) ABOVE BE MOUNTED FOR action? REPLY Y OR N

ARC0366A replies Y to the following WTOR message: ARC0366A REPLY Y ONLY WHEN ALL *nn* TAPE VOLUME(S) IS/ARE COLLECTED...

ARC0381A replies WAIT to the following WTOR message: ARC0381A ALLOC REQ FAILED FOR *volser* FOR action REPLY WAIT OR CANCEL

These Rules are distributed as disabled in Rule Set RULRSRS and must be enabled to implement the solution.

EXECs

There are no EXECs for this solution.

Parameters

The parameters for this solution are in SRSVARG.

HSM Alert Solution

The HSM Alert solution traps messages issued by HSM and issues alerts in response to them.

Prerequisites

Initialization solution

Invocation

This solution is invoked when HSM issues the messages to be trapped.

Rules

This solution uses the following Rules:

ARC0026E issues an alert when the following message is trapped: ARC0026E JOURNALIZING DISABLED DUE TO *error_text*

ARC0036I issues an alert when the following message is trapped: ARC0036I I/O DISABLED FOR DFHSM PROBLEM

ARC0050A issues an alert when the following message is trapped: ARC0050A DFHSM NOT ACTIVE - START DFHSM

ARC0107I issues an alert when the following message is trapped: ARC0107I INSUFFICIENT MAIN STORAGE

ARC0109I issues an alert when the following message is trapped: ARC0109I ACTIVITY LOGGING COULD NOT BE SWITCHED

ARC0305I issues an alert when the following message is trapped: ARC0305I GETMAIN/FREE MAIN FAILURE IN MODULE, *return_code*

ARC0307I issues an alert when the following message is trapped: ARC0307I ERROR MAINING STORAGE FROM SUBPOOL *in mod*

ARC0380A issues an alert when the following message is trapped: ARC0380A RECALL WAITING

ARC0441I issues an alert when the following message is trapped: ARC0441I ALTERNATE VOLUME *alt_volname* REMOVED AS COPY OF VOLUME *volser*

ARC0534I issues an alert when the following message is trapped:
ARC0534I MIGRATION HELD

ARC0560E issues an alert when the following message is trapped:
ARC0560E MIGRATION LIMITED

ARC0057I issues an alert when the following message is trapped:
ARC0057I CSA WSAGE BY DFHSM HAS REACHED THE INACTIVE THRESHOLD OF n BYTES, ALL BUT BATCH WAIT REQUESTS FAILED

ARC0058I issues an alert when the following message is trapped:
ARC0058I CSA WSAGE BY DFHSM HAS REACHED THE ACTIVE THRESHOLD OF *xxxx*, ALL BUT BATCH WAIT REQUESTS FAILED

ARC0708I issues an alert when the following message is trapped:
ARC0708I BKUP|SPILL|CLNUP|DUMP|RSTR TERMINATED

ARC0716I issues an alert when the following message is trapped:
ARC0716I MVMNT BKUP VER|BKUP OF MIGR DSN TERMINATED

ARC0738I issues an alert when the following message is trapped:
ARC0738I FAILURE IN BKUP|DUMP|RCVR|CONTROL TASK, *return_code*

ARC0744E issues an alert when the following message is trapped:
ARC0744E DSID COULD NOT BE BACKED UP, RC = *return_code*, REAS = *reason_code*, MIGRATION BACKUP, DUMP, AND RECYCLE HELD

ARC0860E issues an alert when the following message is trapped:
ARC0860E CDS SPACE MONITORING DISABLED

ARC0909E issues an alert when the following message is trapped:
ARC0909E MIGRATION CONTROL|BACKUP CONTROL|OFFLINE CONTROL|JOURNAL} DATASET IS ABOUT *nn*% FULL

ARC0909I issues an alert when the following message is trapped:
ARC0909I {MIGRATION CONTROL|BACKUP CONTROL|OFFLINE CONTROL|JOURNAL} DATASET IS ABOUT *nn*% FULL

ARC0910E issues an alert when the following message is trapped:
ARC0910E DFHSM CDS IS FULL. REORGANIZE CDS.
MIGRATION, BACKUP, AND DUMP HELD

ARC0923I issues an alert when the following message is trapped:
ARC0923I ERROR CLOSING TAPE DATA SET
dataset_name, return_code

ARC1118I issues an alert when the following message is trapped:
ARC1118I RECALL FAILED ALLOC TAPE

ARC1900I issues an alert when the following message is trapped:
ARC1900I DFHSM ABEND code OCCURRED
PROCESSING *request*

These Rules are distributed as disabled in Rule Set RULRSRS and must be enabled to implement the solution.

EXECs

There are no EXECs for this solution.

Parameters

The parameters for this solution are in SRSVARG.

HSM Message Suppression Solution

The HSM Message Suppression solution traps HSM messages and suppresses them.

Prerequisites

Initialization solution

Invocation

This solution is invoked when HSM issues the messages to be trapped.

Rules

This solution uses the following Rules:

ARC0001I traps and suppresses the following message: ARC0001I DFHSM STARTING

ARC0008I traps and suppresses the following message: ARC0008I DFHSM INITIALIZATION SUCCESSFUL

ARC0020I traps and suppresses the following message: ARC0020I ********************************

ARC0100I traps and suppresses the following message: ARC0100I *xxxxxx* COMMAND COMPLETED

ARC0120I traps and suppresses the following message: ARC0120I PRIMARY VOLUME ADDED

ARC0126I traps and suppresses the following message: ARC0126I ADDROC VOLSER REJECTED

ARC0143I traps and suppresses the following message: ARC0143I PARMLIB MEMBER, *userid, hostid, Primary* HST

ARC0145I traps and suppresses the following message: ARC0145I DS DELETED = DATASETS/FAILS

ARC0146I traps and suppresses the following message: ARC0146I RECYCLED VOLUMES, DS, *Blocks*

ARC0147I traps and suppresses the following message: ARC0147I BUDENSITY, BUUNIT, BU RECYCLE PERCENTAGE

ARC0148I traps and suppresses the following message: ARC0148I TOTAL SPACE, EXTENTS, PERCENT FULL, THRESH

ARC0149I traps and suppresses the following message: ARC0149I MONITOR, MCDS, BCDS, OCDS, JOURNAL THRESH

ARC0150I traps and suppresses the following message: ARC0150J JOURNAL, LOG, TRACE, SMFID, DEBUG, EMERG

ARC0151I traps and suppresses the following message: ARC0151I DAYS, ML1DAYS, PRIMGMT START, MAXTASKS

ARC0152I traps and suppresses the following message: ARC0152I MAXRECALL TASKS, RECALL, PRIV VOLUME, EXTNT

ARC0153I traps and suppresses the following message: ARC0153I SCRATCHFREQ, SYSOUT, SWAP, PERMISSION, EXIT

ARC0154I traps and suppresses the following message: ARC0154I MAXBACKUP TASKS, ABSTRACT, VERSIONS, FREQ

ARC0155I traps and suppresses the following message: ARC0155I DFHSM STATISTICS FOR *date*

ARC0156I traps and suppresses the following message: ARC0156I STARTUPS, SHUTDOWNS, ABENDS, MWES, CPU TIME

ARC0157I traps and suppresses the following message: ARC0157I DS MIGRATE L1/L2, EXTENT REDUCTIONS, FAILS

ARC0158I traps and suppresses the following message: ARC0158I DS RECALL L1/L2, FAILS, RECALL NBYTES

ARC0159I traps and suppresses the following message: ARC0159I DS BACKUP DSN, FAILS, DS RECOVER DSN, FAILS

ARC0164I traps and suppresses the following message: ARC0164I DAY = DAY/SPILL/UNASSIGNED, VOLS

ARC0171I traps and suppresses the following message: ARC0171I SETMIG LEVEL *qualifier* PROCESSED

ARC0175I traps and suppresses the following message: ARC0175I LEVEL *qualifier* AND MIGRATION RESTRICT TYPE

ARC0176I traps and suppresses the following message: ARC0176I *qualifier*, RETRICTION TYPE

ARC0200I traps and suppresses the following message: ARC0200I TRAP IN MODULE *xxx*

ARC0206I traps and suppresses the following message: ARC0206I *address data*

ARC0208I traps and suppresses the following message: ARC0208I TRAP FOR ERROR CODE *xxx*

ARC0210I traps and suppresses the following message: ARC0210I PATCH PROCESSING COMPLETE

ARC0213I traps and suppresses the following message: ARC0213I VERIFY SUCCESSFUL

ARC0216I traps and suppresses the following message: ARC0216I DUMP CLASS DEFINITION, CLASS, RC

ARC0226I traps and suppresses the following message: ARC0226I MIGRATION LEVEL 2 UNDEFINED

ARC0229I traps and suppresses the following message: ARC0229I DATA SET POOL DEFINED

ARC0232I traps and suppresses the following message: ARC0232I DSN/VD2 POOLS, VOLS

ARC0260I traps and suppresses the following message: ARC0260I MIGRATION VOLUME *xxxxxx* ENTRY NOT DEFINED

ARC0270I traps and suppresses the following message: ARC0270I BACKUP CYCLE DEFINITION SUCCESSFUL

ARC0271I traps and suppresses the following message: ARC0271I BACKUP CYCLE LENGTH, CYCLE, DATE, START

ARC0272I traps and suppresses the following message: ARC0272I PRIMARY SPACE MGMT CYCL LENGTH, CYCLE, DATE

ARC0340I traps and suppresses the following message: ARC0340I COMPACTION OPTIONS (TAPE/DASD MIGR, BACKUP)

ARC0341I traps and suppresses the following message: ARC0341I COMPACT PERCENT IS *nn%*

ARC0365I traps and suppresses the following message: ARC0365I VOLUME NOW AVAILABLE FOR RECYCLE

ARC0400I traps and suppresses the following message: ARC0400I VOLUME IS nn% FREE, *trks*, *cyl*, FRAG

ARC0401I traps and suppresses the following message: ARC0401I LARGEST EXTENTS ARE *cyls*, *trks*

ARC0402I traps and suppresses the following message: ARC0402I VTOC IS TRACKS, FREE DSCBS, FORMAT 5 DSCBS

ARC0422I traps and suppresses the following message: ARC0422I TAPECOPY COMPLETED *rc*

ARC0503I traps and suppresses the following message: ARC0503I ALLOCATION ERROR

ARC0517I traps and suppresses the following message: ARC0517I SECONDARY SPACE MGMT STARTING/RESTARTING

ARC0518I traps and suppresses the following message: ARC0518I SECONDARY SPACE MGMT ENDED

ARC0519I traps and suppresses the following message: ARC0519I DADSM PARTEL FUNCTION RELEASED, *trks*, *vol*

ARC0520I traps and suppresses the following message: ARC0520I PRIMARY SPACE MGMT STARTING/RESTARTING

ARC0521I traps and suppresses the following message: ARC0521I PRIMARY SPACE MGMT ENDED SUCCESSFULLY

ARC0522I traps and suppresses the following message: ARC0522I SPACE MANAGEMENT STARTING ON VOLUME *xxxxxx*

ARC0523I traps and suppresses the following message: ARC0523I SPACE MANAGEMENT ENDED ON VOLUME *xxxxxx*

ARC0526I traps and suppresses the following message: ARC0526I MIGR CLEANUP STARTING AT *time*, *date*, *system*

ARC0527I traps and suppresses the following message: ARC0527I MIGR CLEANUP ENDED AT *time*, *date*, *system*

ARC0529I traps and suppresses the following message: ARC0529I MIGR CLEANUP PARAMETERS FOR *recs*, *days*, *old*

ARC0718I traps and suppresses the following message: ARC0718I MOVEMENT OF BKUP VERSIONS STARTNG

ARC0719I traps and suppresses the following message: ARC0719I MOVEMENT OF BKUP VERSIONS ENDING

ARC0720I traps and suppresses the following message: ARC0720I AUTOMATIC BACKUP STARTING

ARC0721I traps and suppresses the following message: ARC0721I AUTOMATIC BACKUP ENDING

ARC0722I traps and suppresses the following message: ARC0722I BACKUP STARTING ON VOLUME xxxxxx AT *hh:mm:ss*

ARC0723I traps and suppresses the following message: ARC0723I BACKUP ENDING ON VOLUME *xxxxxx* AT *hh:mm:ss*

ARC0724I traps and suppresses the following message: ARC0724I CLEANUP STARTING ON BACKUP VOLUME *xxxxxx*

ARC0725I traps and suppresses the following message: ARC0725I BACKUP SPILL IS USING VOLUME *xxxxxx*

ARC0726I traps and suppresses the following message: ARC0726I CLEANUP ENDING ON BACKUP VOLUME *xxxxxx*

ARC0728I traps and suppresses the following message: ARC0728I VTOC FOR VOL *vvvvvv* COPIED TO DATA SET

ARC0734I traps and suppresses the following message: ARC0734I ACTION=*xxx*,DSN=*xxx*,ACTION=*xxxx*,FRVOL=*xxx*,TOVOL=*xxx*, TRACKS...

ARC0735I traps and suppresses the following message: ARC0735I BKUP OF MIGRATED DS STARTING

ARC0736I traps and suppresses the following message: ARC0736I BKUP OF MIGRATED DS ENDING

ARC0740I traps and suppresses the following message: ARC0740I CDS BACKUP STARTING AT *time* ON *date* SYSTEM *sysid*, TO TAPE/DASD IN PARALLELL/NONPARALLEL MODE

ARC0741I traps and suppresses the following message: ARC0741I CDS BACKUP ENDING AT *time* ON *date*, STATUS = *status*

ARC0742I traps and suppresses the following message: ARC0742I BACKUP FOR XCDS STARTING

ARC0743I traps and suppresses the following message: ARC0743I XCDS SUCCESSFULLY BACKED UP

ARC0748I traps and suppresses the following message: ARC0748I LAST SUCCESSFULL CDS BKUP QUALIF

ARC0831I traps and suppresses the following message: ARC0831I RECYCLE COMMAND PROCESSING ENDED

ARC0832I traps and suppresses the following message: ARC0832I RECYCLE STARTING ON VOLUME *xxxxxx* AT *hh:mm:ss*

ARC0833I traps and suppresses the following message: ARC0833I RECYCLE ENDED ON VOLUME *xxxxxx* AT *hh:mm:ss*

ARC1800I traps and suppresses the following message: ARC1800I DFHSM WAIT REQUEST CANCELLED BY ATTENTION

These Rules are distributed as disabled in Rule Set RULRSRS and must be enabled to implement the solution.

EXECs

There are no EXECs for this solution.

Parameters

The parameters for this solution are in SRSVARG.

ADSM Hung Session Solution

The ADSM Hung Session solution issues the ADSM QUERY SESSION command to check for outstanding ADSM sessions. When it finds an outstanding session older than the user-specified threshold, the session is either canceled or an alert is issued.

Prerequisites

Initialization solution

Invocation

This solution is invoked by a time-initiated Rule. It can also be invoked manually by issuing the following AutoOPERATOR console command:

```
%ADSMHUNG
```

Rules

This solution uses the following Rule:

ADSMHUNG time-initiated Rule that executes every hour at four minutes past the hour

If you want to change the interval, use the AutoOPERATOR Rules application to change it.

This Rule is distributed as enabled in Rule Set RULRSRS.

EXECs

This solution uses the following EXEC:

ADSMHUNG issues the ADSM QUERY SESSION command and parses through the output looking for an elapsed time value over the user-specified limit

Each time ADSMHUNG finds an elapsed time over the user-specified limit, an alert is issued (if in test mode) or the session is cancelled.

ADSMWTOR issues a write-to-operator-with-reply (WTOR) message with instructions to cancel the session

Parameters

The parameters for this solution are in parmlib member SRSVARD. The parameters are read by the Initialization solution and saved as shared variables.

Parameters	Descriptions
ADSMHUNG_Debug	sets verbose mode for the solution 0 = off (normal) 1 = verbose 2 = very verbose.
ADSM_Alert_Priority	priority for all alerts created by this solution for duplicate requests; values are CRITICAL, MAJOR, MINOR, WARN, INFO, or CLEARING.
ADSM_Test	sets test mode 0= off 1 = on When off, this solution will issue an alert instead of canceling the duplicate request.
ADSMHUNG_Threshold	threshold value in minutes If a session has been active for longer than this number it will be cancelled.

Appendix A System Event Definitions

The following system event definitions cannot be changed by the user:

SVWI0010E: EVENT *event_id* NOT FOUND IN EVENT DEFINITIONS, REQUESTED BY FUNCTION *function_name*

Explanation: This event is generated by Enterprise Storage Automation when a user event ID specified in an FLST/RLST member cannot be located in the active SMEVNT*xx* member. This event indicates that the event in error is not defined in SMEVNT*xx* or that the wrong SMEVNT*xx* member is active.

SVWI1001W *** WARNING *** *appl_name appl_type* IS AT *pct* % OF BUDGETED SPACE (CUR=*cur* MAX=*max*

Explanation: This event is generated through SG-Control when a space allocation request has caused the warning threshold to be exceeded. This event will not be generated if SG-Control is running in MONITOR mode.

Events cannot be generated from SG-Control if the SG-Control user exit facility is used and issues a return code of RC4 or RC8. A return code of RC4 or RC8 indicates that SG-Control budget messages will be bypassed.

SVWI1002W ACCOUNT *appl_name appl_type sys_ action alloc_amt* (CUR=*cur* MAX=*max*)

Explanation: This event is generated when the budgeted amount of space for the application (*appl_name* and *appl_type*) is exceeded. The amount requested (*alloc_amt*) plus the current amount used (*cur*) is greater than the budget (*max*) allows for the type of storage specified in the message. This event will not generated if SG-Control is running in MONITOR mode.

Appendix B REXX EXECs

The following REXX EXECs are required when submitting jobs using console commands or the ACT_JOB keyword in any AUTO function RLST member:

- CORSR005—allows the submission of a job using a console command
- CORSR015— notifies SVOS when a job submitted through ACT_JOB keyword has ended
- CORSR020—deletes variables according to a user-defined time frame

The REXX EXECs described in this appendix are

CORSR005 – Job Submission Using Console Commands

Submitting a job can be an automation solution. You can submit a job using a console command, which invokes REXX EXEC CORSR005, or you can create an event and an AutoOPERATOR Rule definition that passes parameters to the REXX EXEC CORSR005. CORSR005 processes the parameters that are passed as variables to build the JCL and submit the batch job. The parameters passed are converted to variables placed in array format using stem variable names. The variables shown in Table B-1 are used for substitution as part of the skeleton tailoring process in the building of the JCL to be submitted.

Table B-1 Input/Output Variables for CORSR005

Input Variables Used on Invocation of REXX EXEC	Output Variables Used in Skeleton JCL
A= or APPL=	APPL.*n*
DS= or DSN=	DSN.*n*
DV= or DEVT=	DEVT.*n*
G= or GROUP=	GROUP.*n*
J= or JOB=	JOB.*n*
P= or POOL=	POOL.*n*
RT= or RESTYPE=	RESTYPE.*n*
F= or FUNC=	FUNC.*n*
S= or SOL=	SOL.*n*
V= or VOLSER=	VOLSER.*n*
**CORSR005_DEBUG	Messages to the AutoOPERATOR Journal

Required Parameter

The J= or JOB= parameter must contain the name of a member in a PDS library that is concatenated in the SYSPROC DD statement in the AutoOPERATOR procedure. This member contains the skeleton JCL to be used during the variable substitution process. Not coding the J= or JOB= parameter will cause execution to stop.

Related Information

Only one parameter of each type is allowed. Therefore, either P= or POOL= can be used, either V= or VOLSER= can be used, and so on. Coding duplicate parameters will cause execution to stop.

P= or POOL= and G= or GROUP= are mutually exclusive. However, coding either will cause the other to be created. So, if P=*mypool* is coded then, a variable POOL.*n* is created with the value of *mypool* and a variable GROUP.*n* is also created with the value of *mypool*.

This REXX EXEC will accept the first six parameters coded. Any parameters following the sixth parameter are ignored.

** CORSR005_DEBUG parameter is located in the SRSVARJ member located in your AutoOPERATOR parmlib. The default value is CORSR005_DEBUG=0, indicating that no informational messages will be written to the AutoOPERATOR Journal. Setting this value to 1 will cause messages to be written to the AutoOPERATOR Journal. To update this parameter, you must edit the SRSVARJ member and issue the following AutoOPERATOR console command:

```
%SRSVAR SRSVARJ
```

Writing an excessive number of messages to the AutoOPERATOR Journal can impede performance. It is recommended that the default value not be changed unless instructed to do so by BMC Software Customer Support.

Examples

The following examples show variable substitutions and ways to invoke CORSR005, which builds the JCL and submits a batch job. For an explanation of skeleton tailoring, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

Example

Invoke CORSR005 through a modify command:

```
/F AOSTC,%CORSR005 J=TSTJCL1 V=VOL001  DS=TEST.JCL
```

where AOSTC is your AutoOPERATOR Started Task name

OR

Example

Invoke CORSR005 using MAINVIEW SRM Events and AutoOPERATOR Rule definitions.

Event text—defined in an event member in SMMSYS*xx*:

```
SET EVENTID=SK001
SEV=I
MODE=ACT
TEXT='DSN &AD_DSN on volume &AD_VOL not cataloged'
```

Replacement text—event text as it would be replaced during event processing:

SVWSK001I *function STC sysid* DSN TEST.JCL on volume VOL001 not cataloged

AutoOPERATOR Rule—defined in the “Action Specification” panel in AutoOPERATOR for the Rule that fires on eventid SVWSK001I:

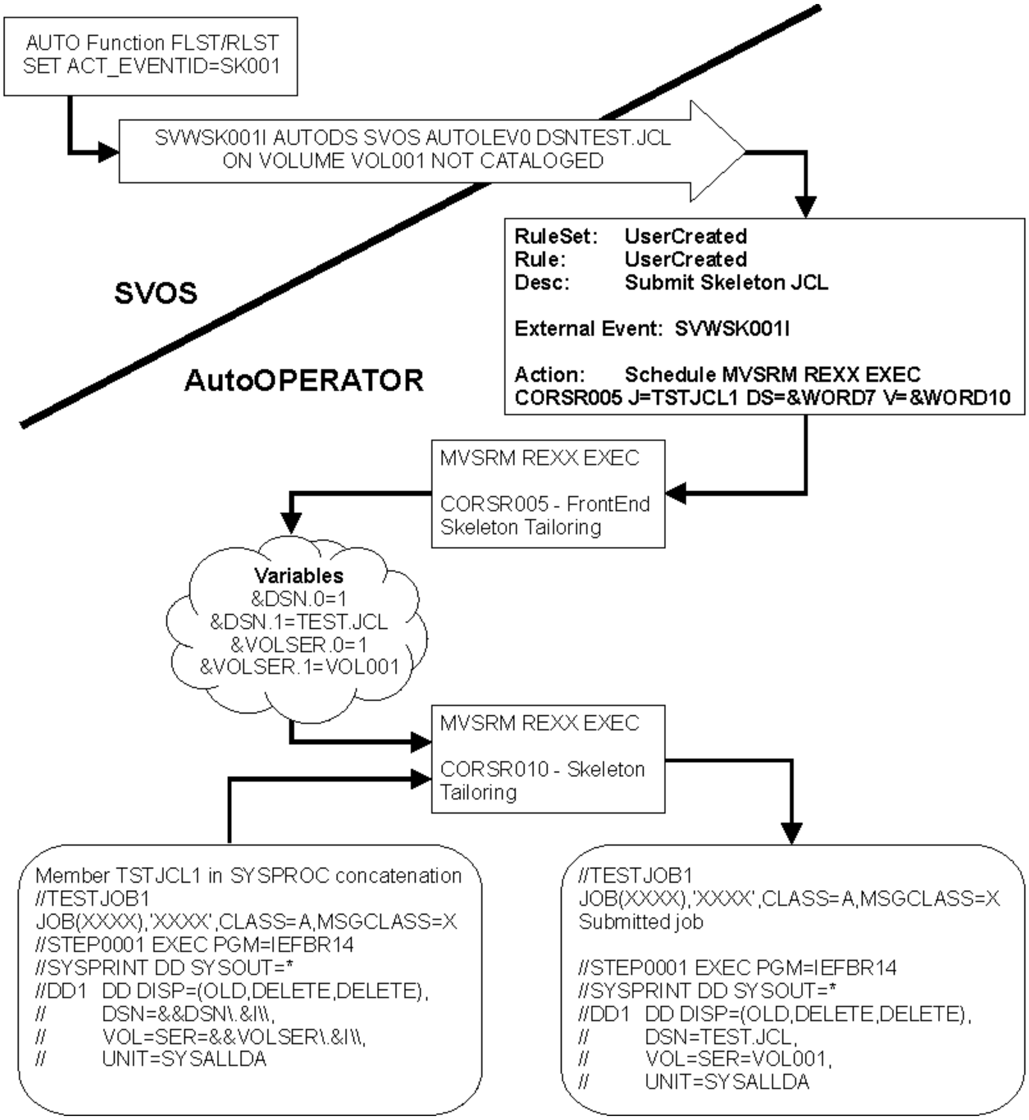
```
EXEC Name/Parms  ===> CORSR005 J=TSTJCL1 DS=&WORD3
V=&WORD6
```

The “&WORD*n*” correlates to the word position in the Event text of the desired variable replacement.

Skeleton JCL—JCL that is located in a member named TSTJCL1 in the SYSPROC DD concatenation:

```
//TESTJOB1 JOB (XXXX),’XXXX’,CLASS=A,MSGCLASS=X
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1  DD DISP=(OLD,DELETE,DELETE),
//          DSN=&&DSN\.&I\\,
//          VOL=SER=&&VOLSER\.&I\\,
//          UNIT=SYSALLDA
```

Figure B-1 Job Submission from an Event Diagram


AUTO Function FLST/RLST
SET ACT_EVENTID=SK001
SVWSK001I AUTODS SVOS AUTOLEV0 DSNTEST.JCL
ON VOLUME VOL001 NOT CATALOGED
SVOS
AutoOPERATOR
RuleSet: UserCreated
Rule: UserCreated
Desc: Submit Skeleton JCL
External Event: SVWSK001I
Action: Schedule MVSRM REXX EXEC
CORSR005 J=TSTJCL1 DS=&WORD7 V=&WORD10
MVSRM REXX EXEC
CORSR005 - FrontEnd Skeleton Tailoring
Variables
&DSN.0=1
&DSN.1=TEST.JCL
&VOLSER.0=1
&VOLSER.1=VOL001
MVSRM REXX EXEC
CORSR010 - Skeleton Tailoring

Member TSTJCL1 in SYSPROC concatenation

```
//TESTJOB1
JOB(XXXX),'XXXX',CLASS=A,MSGCLASS=X
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1   DD DISP=(OLD,DELETE,DELETE),
//         DSN=&&DSN\.&I\\,
//         VOL=SER=&&VOLSER\.&I\\,
//         UNIT=SYSALLDA
```

```
//TESTJOB1
JOB(XXXX),'XXXX',CLASS=A,MSGCLASS=X
```

Submitted job

```
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1   DD DISP=(OLD,DELETE,DELETE),
//         DSN=TEST.JCL,
//         VOL=SER=VOL001,
//         UNIT=SYSALLDA
```

Example

Either invocation produces the following result using the TSTJCL1 skeleton member:

```
//TESTJOB1 JOB (XXXX),’XXXX’,CLASS=A,MSGCLASS=X
//STEP0001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=*
//DD1  DD DISP=(OLD,DELETE,DELETE),
//          DSN=TEST.JCL,
//          VOL=SER=VOL001,
//          UNIT=SYSALLDA
```

CORSR015 – Job Submission Using ACT_JOB Keyword

Enterprise Storage Automation keeps track of outstanding jobs submitted through the AUTO function RLST. Enterprise Storage Automation must be notified when a submitted job ends. This REXX EXEC performs this notification, using variables that were established when the job was submitted by the AUTO function.

If the ACT_JOB keyword is used in the AUTO functions RLST, the Rules distributed in Rule Set RULSRS01 must be enabled. These Rules fire base on MVS and JES messages issued to the console log signaling a job ending. The Rules then invoke CORSR015 to perform the job end notification to Enterprise Storage Automation.

Warning! The Rules in this Rule Set filter not only on the message number but also on the job name. This filter is required. Not using this filter could cause sever performance degradation within AutoOPERATOR, to the point of failure.

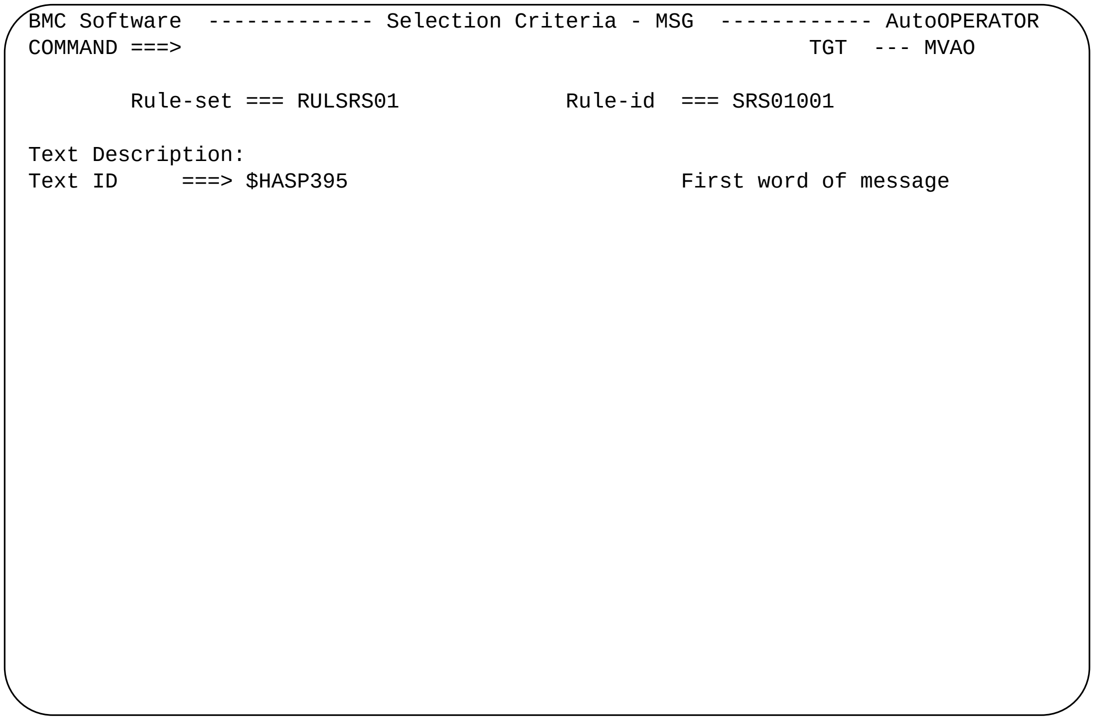
All jobs submitted through Enterprise Storage Automation should either 1) all contain a common jobname prefix, which can then be specified in the Rules in RULSRS01 Rule Set or, 2) each specific job name should be specified in the RULSRS01 Rule Set. The requirement is that an AutoOPERATOR Rule fire at job end, and that Rule must filter on jobname.

Required Parameters: None

Example

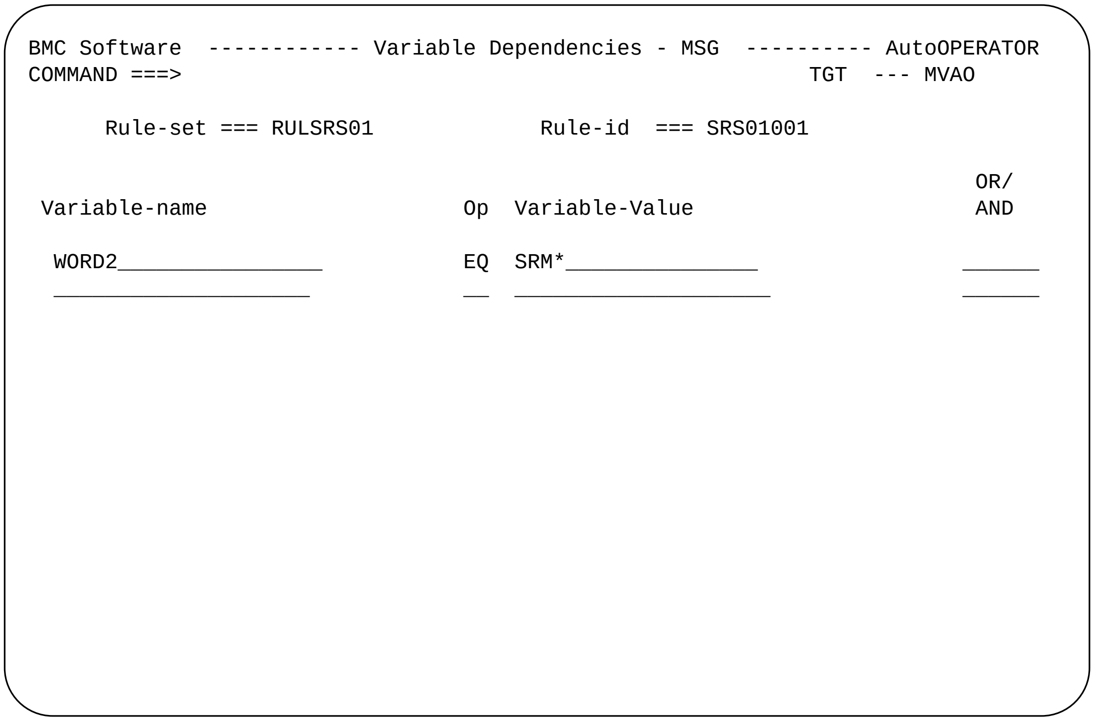
Rule as defined in the **Selection Criteria** panel of AutoOPERATOR:

Figure B-2 RULSRS01 Rule ID SRS01001 as defined in the Selection Criteria panel of AutoOPERATOR



```
BMC Software  ------------- Selection Criteria - MSG  ------------ AutoOPERATOR
COMMAND ===>                                                 TGT  --- MVAO

        Rule-set === RULSRS01             Rule-id  === SRS01001

Text Description:
Text ID     ===> $HASP395                          First word of message
```

Figure B-3 **RULSRS01 Rule ID SRS01001 as defined in the Variable Dependencies panel of AutoOPERATOR**

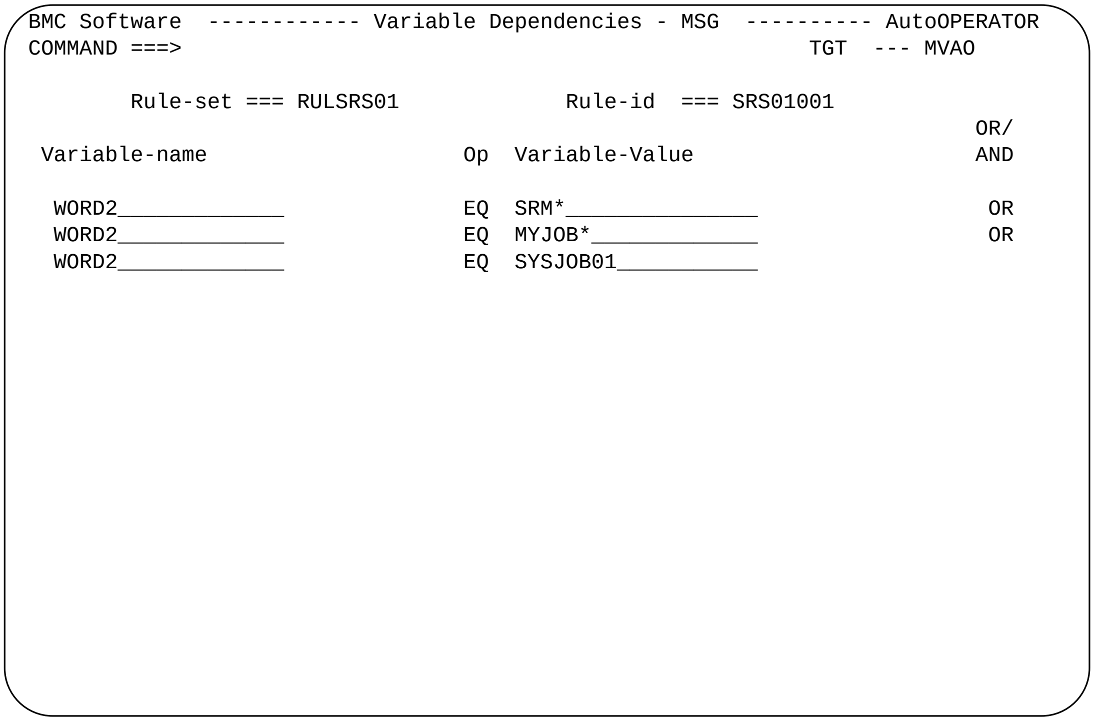


```
BMC Software  ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                 TGT  --- MVAO

      Rule-set === RULSRS01            Rule-id  === SRS01001

                                                                          OR/
 Variable-name                     Op  Variable-Value                     AND

  WORD2________________            EQ  SRM*_______________               ______
  ____________________             __  ____________________              ______
```

Where WORD2 is the job name (the second word) in the $HASP395 message. In this example it is matching on any SRM prefixed job name.

If you are using more than one job name prefix you can expand this definition by using the OR parameter under the AND/OR column. See the *MAINVIEW AutoOPERATOR Basic Automation Guide* for more information.

Figure B-4 **RULSRS01 Rule ID SRS01001 as defined in the Variable Dependencies panel of AutoOPERATOR (Extended example)**



```
BMC Software  ------------ Variable Dependencies - MSG  ---------- AutoOPERATOR
COMMAND ===>                                                TGT  --- MVAO

        Rule-set === RULSRS01            Rule-id  === SRS01001
                                                                        OR/
 Variable-name                   Op  Variable-Value                     AND

  WORD2_____________             EQ  SRM*_______________                 OR
  WORD2_____________             EQ  MYJOB*_____________                 OR
  WORD2_____________             EQ  SYSJOB01___________
```

Currently there are three Rule IDs in the RULSRS01 Rule Set. Each filtering on a different possible MVS or JES job end message. The Variable Dependencies for each Rule ID *must* be updated to reflect the job naming conventions used in the skeleton JCL members.

Example

Rule as defined in the **Selection Criteria** panel of AutoOPERATOR for previous examples:

EXEC Name/Parms ===> **CORSR015**

Table B-2 **Input/Output Variables for CORSR015**

Input Variables Used on Invocation of REXX EXEC	Output Variables Used in Skeleton JCL
**CORSR015_DEBUG	Messages to the AutoOPERATOR Journal

** CORSR015_DEBUG parameter is located in the SRSVARJ member located in your AutoOPERATOR parmlib. The default value for CORSR015_DEBUG=0 indicating no informational messages will be written to the AutoOPERATOR Journal. Setting this value to 1 will cause messages to be written to the AutoOPERATOR Journal. To update this parameter, you must edit the SRSVARJ member and issue the following AutoOPERATOR console command:

```
%SRSVAR SRSVARJ.
```

Writing excessive number of messages to the AutoOPERATOR Journal can impede performance. It is recommended that the default value not be changed unless instructed to do so by BMC Software Customer Support.

CORSR020 – Delete Variables

In the Enterprise Storage Automation job submission process, a number JOB END command variables are created within AutoOPERATOR for each submitted job. Under normal circumstances, these variables are utilized (the commands issued) and then deleted when the job ends. In the event a disruption occurs, it is possible that the commands may not be issued and that the variables may not get deleted from the AutoOPERATOR shared pool.

The function of CORSR020 is to issue the commands and then delete these variables within a user-defined time frame. CORSR020 can be invoked from the console as a command or as a time-initiated AutoOPERATOR Rule.

Required Parameters: ***n*DAYS or *n*HOURS or ALL**

Example

```
/F AOSTC,%CORSR020
```

```
/F AOSTC,%CORSR020 7HOURS
```

where AOSTC is the AutoOPERATOR Started Task Name

In the first example, the time duration will be taken from the *MVSRM_CMDVAR_EXP variable in the AutoOPERATOR shared variable pool.

In the second example, any command variables (MVSRM.jnum.CMD.*n*) that have been in the shared pool for 7 hours or more will be issued and then deleted. By passing a duration when the CORSR020 REXX EXEC is invoked you override the MVSRM_CMDVAR_EXP variable value.

Warning! Before using REXX EXEC, you need an understanding of automation processing and the expected duration of the processing any time you use ACT_JOB to submit a job. Passing a duration time to CORSR020 that is shorter than the expected processing time could result in disruption of automation processing.

For example, if a cycle starts at 10 A.M. and the complete cycle ends (all jobs completed) at 10 P.M., running CORSR020 during that time period passing a parameter of 3HOURS could cause a disruption of the automation processing cycle.

Table B-3 Input Variables for CORSR020

Input Variables Used on Invocation of REXX EXEC	Description
nDAYS	number of days since the variables were created
*n*HOURS	number of hours since the variables were created
ALL	issues all the job end commands and deletes all the variables; not valid on for use with the MVSRM_CMDVAR_EXP parameter
*MVSRM_CMDVAR_EXP=nDAYS	default duration placed in the *n*HOURS AutoOPERATOR shared variable pool by SRSVAR command
**CORSR020_DEBUG	Messages to the AutoOPERATOR Journal

*MVSRM_CMDVAR_EXP is updated using the SRSVARJ member and the %SRSVAR SRSVARJ command. The default value for MVSRM_CMDVAR_EXP is 9HOURS, indicating that any MVSRM.*jnum*.CMD.*n* variables will be issued and then deleted if they have been in the shared pool for 9 or more hours.

** CORSR020_DEBUG parameters are located in the SRSVARJ member located in your AutoOPERATOR parmlib. The default value is CORSR020_DEBUG=0 indicating no informational messages will be written to the AutoOPERATOR Journal. Setting this value to 1 will cause messages to be written to the AutoOPERATOR Journal. To update these parameters you must edit the SRSVARJ member and issue the following AutoOPERATOR console command:

```
%SRSVAR SRSVARJ
```

Writing excessive number of messages to the AutoOPERATOR Journal can impede performance. It is recommended that the default value not be changed unless instructed to do so by BMC Software Customer Support.

Glossary

This glossary defines BMC Software terminology. Other dictionaries and glossaries can be used in conjunction with this glossary.

Since this glossary pertains to BMC Software-related products, some of the terms defined might not appear in this book.

To help you find the information you need, this glossary uses the following cross-references:

Contrast with indicates a term that has a contrary or contradictory meaning.

See indicates an entry that is a synonym or contains expanded information.

See also indicates an entry that contains related information.

action Defined operation, such as modifying a MAINVIEW window, that is performed in response to a command. *See* object.

active window Any MAINVIEW window in which data can be refreshed. *See* alternate window, current window, window.

administrative view Display from which a product's management tasks are performed, such as the DSLIST view for managing historical data sets. *See* view.

ALT WIN field Input field that allows you to specify the window identifier for an alternate window where the results of a hyperlink are displayed. *See* alternate window.

Alternate Access *See* MAINVIEW Alternate Access.

alternate form View requested through the FORM command that changes the format of a previously displayed view to show related information. *See also* form, query.

alternate window (1) Window that is specifically selected to display the results of a hyperlink. (2) Window whose identifier is defined to the ALT WIN field. *Contrast with* current window. *See* active window, window, ALT WIN field.

analyzer (1) Online display that presents a snapshot of status and activity data and indicates problem areas. (2) Component of CMF MONITOR. *See* CMF MONITOR Analyzer.

application (1) Program that performs a specific set of tasks within a MAINVIEW product. (2) In MAINVIEW VistaPoint, combination of workloads to enable display of their transaction performance data in a single view.

application trace *See* trace.

ASCH workload Workload comprising Advanced Program-to-Program Communication (APPC) address spaces.

AutoCustomization Online facility for customizing the installation of products. AutoCustomization provides an ISPF panel interface that both presents customization steps in sequence and provides current status information about the progress of the installation.

automatic screen update

Usage mode wherein the currently displayed screen is refreshed automatically with new data at an interval you specify. Invoked by the ASU command.

batch workload Workload consisting of address spaces running batch jobs.

BBI Basic architecture that distributes work between workstations and multiple OS/390 targets for BMC Software MAINVIEW products.

BBI-SS PAS *See* BBI subsystem product address space.

BBI subsystem product address space (BBI-SS PAS)

OS/390 subsystem address space that manages communication between local and remote systems and that contains one or more of the following products:

- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for DB2
- MAINVIEW for DBCTL
- MAINVIEW for IMS Online
- MAINVIEW for MQSeries (formerly Command MQ for S/390)
- MAINVIEW SRM
- MAINVIEW VistaPoint (for CICS, DB2, DBCTl, and IMS workloads)

BBPARM *See* parameter library.

BBPROC *See* procedure library.

BBPROF *See* profile library.

BBSAMP *See* sample library.

BBV *See* MAINVIEW Alternate Access.

BBXS BMC Software Subsystem Services. Common set of service routines loaded into common storage and used by several BMC Software MAINVIEW products.

border Visual indication of the boundaries of a window.

bottleneck analysis Process of determining which resources have insufficient capacity to provide acceptable service levels and that therefore can cause performance problems.

CA-Disk Data management system by Computer Associates that replaced the DMS product.

CAS Coordinating address space. One of the address spaces used by the MAINVIEW windows environment architecture. The CAS supplies common services and enables communication between linked systems. Each OS/390 or z/OS image requires a separate CAS. Cross-system communication is established through the CAS using VTAM and XCF communication links.

CFMON *See* coupling facility monitoring.

chart Display format for graphical data. *See also* graph.

CICSplex User-defined set of one or more CICS systems that are controlled and managed as a single functional entity.

CMF MONITOR Comprehensive Management Facility MONITOR. Product that measures and reports on all critical system resources, such as CPU, channel, and device usage; memory, paging, and swapping activity; and workload performance.

CMF MONITOR Analyzer
Batch component of CMF MONITOR that reads the SMF user and 70 series records created by the CMF MONITOR Extractor and/or the RMF Extractor and formats them into printed system performance reports.

CMF MONITOR Extractor
Component of CMF that collects performance statistics for CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390, and RMF postprocessor. *See* CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390.

CMF MONITOR Online

Component of CMF that uses the MAINVIEW window interface to present data on all address spaces, their use of various system resources, and the delays that each address space incurs while waiting for access to these resources. *See* CMF MONITOR, MAINVIEW for OS/390.

CMF Type 79 API Application programming interface, provided by CMF, that provides access to MAINVIEW SMF-type 79 records.

CMFMON Component of CMF MONITOR that simplifies online retrieval of information about system hardware and application performance and creates MAINVIEW SMF-type 79 records.

The CMFMON *online facility* can be used to view data in one or more formatted screens.

The CMFMON *write facility* can be used to write collected data as MAINVIEW SMF-type 79 records to an SMF or sequential data set.

CMRDETL MAINVIEW for CICS data set that stores detail transaction records (type 6E) and abend records (type 6D). Detail records are logged for each successful transaction. Abend records are written when an abend occurs. Both records have the same format when stored on CMRDETL.

CMRSTATS MAINVIEW for CICS data set that stores both CICS operational statistic records, at five-minute intervals, and other records, at intervals defined by parameters specified during customization (using CMRSOPT).

column Vertical component of a view or display, typically containing fields of the same type of information, that varies by the objects associated in each row.

collection interval Length of time data is collected. *See also* delta mode, total mode.

command delimiter Special character, usually a **;** (semicolon), used to stack commands typed concurrently on the COMMAND line for sequential execution.

COMMAND line Line in the control area of the display screen where primary commands can be typed. *Contrast with* line command column.

Command MQ Automation D/S

Command MQ agents, which provide local proactive monitoring for both MQSeries and MSMQ (Microsoft message queue manager). The Command MQ agents operate at the local node level where they continue to perform functions regardless of the availability of the MQM (message queue manager) network. Functionality includes automatic monitoring and restarts of channels, queue managers, queues and command servers. In cases where automated recovery is not possible, the agents transport critical alert information to a central console.

Command MQ Automation S/390

Command MQ component, which monitors the MQM (message queue manager) networks and intercedes to perform corrective actions when problems arise. Solutions include:

- Dead-Letter Queue management
- System Queue Archival
- Service Interval Performance solutions
- Channel Availability

These solutions help ensure immediate relief to some of the most pressing MQM operations and performance problems.

Command MQ for D/S

Command MQ for D/S utilizes a true client/server architecture and employs resident agents to provide configuration, administration, performance monitoring and operations management for the MQM (message queue manager) network.

Command MQ for S/390

See MAINVIEW for MQSeries.

COMMON STORAGE MONITOR

Component of MAINVIEW for OS/390 that monitors usage and reconfigures OS/390 or z/OS common storage blocks.

composite workload

Workload made up of a WLM workload or other workloads, which are called *constituent workloads.*

constituent workload

Member of a composite workload. Constituent workloads in a composite usually belong to a single workload class, but sometimes are mixed.

contention Occurs when there are more requests for service than there are servers available.

context In a Plex Manager view, field that contains the name of a target or group of targets specified with the CONTEXT command. *See* scope, service point, SSI context, target context.

CONTEXT command

Specifies either a MAINVIEW product and a specific target for that product (*see* target context) or a MAINVIEW product and a name representing one or more targets (*see* SSI context) for that product.

control statement (1) Statement that interrupts a sequence of instructions and transfers control to another part of the program. (2) Statement that names samplers and other parameters that configure the MAINVIEW components to perform specified functions. (3) In CMF MONITOR, statement in a parameter library member used to identify a sampler in the extractor or a report in the analyzer, or to describe either component’s processing requirements to the operating system.

coupling facility monitoring (CFMON)
Coupling facility views that monitor the activity of your system’s coupling facilities.

current data Data that reflects the system in its current state. The two types of current data are real-time data and interval data. *Contrast with* historical data. *See also* interval data, real-time data.

current window In the MAINVIEW window environment, window where the main dialog with the application takes place. The current window is used as the default window destination for commands issued on the COMMAND line when no window number is specified. *Contrast with* alternate window. *See* active window, window.

DASD (Direct Access Storage Device) (1) A device with rotating recording surfaces that provides immediate access to stored data. (2) Any device that responds to a DASD program.

DASD ADVISOR An interactive software tool that diagnoses DASD performance problems and makes recommendations to reduce overall service time. This tool measures and reports on the operational performance of IBM and IBM-compatible devices.

data collector Program that belongs to a MAINVIEW product and that collects data from various sources and stores the data in records used by views. For example, MAINVIEW for OS/390 data collectors obtain data from OS/390 or z/OS services, OS/390 or z/OS control blocks, CMF MONITOR Extractor control blocks, and other sources. *Contrast with* extractor.

delta mode (1) In MAINVIEW for DB2 analyzer displays, difference between the value sampled at the start of the current statistics interval and the value sampled by the current analyzer request. *See also* statistics interval. (2) In CMFMON, usage mode wherein certain columns of data reflect the difference in values between one sample cycle and the next. Invoked by the DELta ON command. *See also* collection interval, sample cycle, total mode.

DFSMS (Data Facility Storage Management System) Data management, backup, and HSM software from IBM for OS/390 or z/OS mainframes.

DMR *See* MAINVIEW for DB2.

DMS (Data Management System) *See* CA-Disk.

DMS2HSM *See* MAINVIEW SRM DMS2HSM.

DSO (Data Set Optimizer) CMF MONITOR Extractor component that uses CMF MONITOR Extractor data to produce reports specifying the optimal ordering of data sets on moveable head devices.

EasyHSM *See* MAINVIEW SRM EasyHSM.

EasyPOOL *See* MAINVIEW SRM EasyPOOL.

EasySMS *See* MAINVIEW SRM EasySMS.

element (1) Data component of a data collector record, shown in a view as a field. (2) Internal value of a field in a view, used in product functions.

element help Online help for a field in a view. The preferred term is *field help*.

Enterprise Storage Automation
See MAINVIEW SRM Enterprise Storage Automation.

event A message issued by Enterprise Storage Automation. User-defined storage occurrences generate events in the form of messages. These events provide an early warning system for storage problems and are routed to user-specified destinations for central viewing and management.

Event Collector Component for MAINVIEW for IMS Online, MAINVIEW for IMS Offline, and MAINVIEW for DBCTL that collects data about events in the IMS environment. This data is required for Workload Monitor and optional for Workload Analyzer (except for the workload trace service). This data also is recorded as transaction records (X‘FA’) and program records (X‘F9’) on the IMS system log for later use by the MAINVIEW for IMS Offline components: Performance Reporter and Transaction Accountant.

expand Predefined link from one display to a related display. *See also* hyperlink.

extractor Program that collects data from various sources and keeps the data control blocks to be written as records. Extractors obtain data from services, control blocks, and other sources. *Contrast with* data collector.

extractor interval *See* collection interval.

fast path Predefined link between one screen and another. To use the fast path, place the cursor on a single value in a field and press **Enter**. The resulting screen displays more detailed information about the selected value. *See also* hyperlink.

field Group of character positions within a screen or report used to type or display specific information.

field help Online help describing the purpose or contents of a field on a screen. To display field help, place the cursor anywhere in a field and press **PF1** (HELP). In some products, field help is accessible from the screen help that is displayed when you press **PF1**.

filter Selection criteria used to limit the number of rows displayed in a view. Data that does not meet the selection criteria is not displayed. A filter is composed of an element, an operator, and an operand (a number or character string). Filters can be implemented in view customization, through the PARm/QPARm commands, or through the Where/QWhere commands. Filters are established against elements of data.

fire The term used to indicate that an event has triggered an action. In MAINVIEW AutoOPERATOR, when a rule selection criteria matches an incoming event and *fires*, the user-specified automation actions are performed. This process is also called *handling* the event.

fixed field Field that remains stationary at the left margin of a screen that is scrolled either right or left.

FOCAL POINT MAINVIEW product that displays a summary of key performance indicators across systems, sites, and applications from a single terminal.

form One of two constituent parts of a view; the other is query. A form defines how the data is presented; a query identifies the data required for the view. *See also* query, view.

full-screen mode Display of a MAINVIEW product application or service on the entire screen. There is no window information line. *Contrast with* windows mode.

global command Any MAINVIEW window interface command that can affect all windows in the window area of a MAINVIEW display.

graph Graphical display of data that you select from a MAINVIEW window environment view. *See also* chart.

hilevel For MAINVIEW products, high-level data set qualifier required by a site’s naming conventions.

historical data (1) Data that reflects the system as it existed at the end of a past recording interval or the duration of several intervals. (2) Any data stored in the historical database and retrieved using the TIME command. *Contrast with* current data, interval data and real-time data.

historical database Collection of performance data written at the end of each installation-defined recording interval and containing up to 100 VSAM clusters. Data is extracted from the historical database with the TIME command. *See* historical data.

historical data set In MAINVIEW products that display historical data, VSAM cluster file in which data is recorded at regular intervals.

HSM (Hierarchical Storage Management) Automatic movement of files from hard disk to slower, less-expensive storage media. The typical hierarchy is from magnetic disk to optical disk to tape.

hyperlink (1) Preset field in a view or an EXPAND line on a display that permits you to

- access cursor-sensitive help
- issue commands
- link to another view or display

The transfer can be either within a single product or to a related display/view in a different BMC Software product. Generally, hyperlinked fields are highlighted. (2) Cursor-activated short path from a topic or term in online help to related information. *See also* fast path.

Image log Collection of screen-display records. Image logs can be created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Image log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Image log stops when both data sets are filled and the first data set is not processed by the archive program.

The TS Image log is a single data set that wraps around when full.

IMSPlex System Manager (IPSM)
MVIMS Online and MVDBC service that provides Single System Image views of resources and bottlenecks for applications across one or more IMS regions and systems.

interval data Cumulative data collected during a collection interval. Intervals usually last from 15 to 30 minutes depending on how the recording interval is specified during product customization. *Contrast with* historical data.

Note: If change is made to the workloads, a new interval will be started.

See also current data and real-time data.

InTune Product for improving application program performance. It monitors the program and provides information used to reduce bottlenecks and delays.

IRUF

IMS Resource Utilization File (IRUF). IRUFs can be either detail (one event, one record) or summarized (more than one event, one record). A detail IRUF is created by processing the IMS system log through a program called IMFLEDIT. A summarized IRUF is created by processing one or more detail IRUFs, one or more summarized IRUFs, or a combination of both, through a sort program and the TASCOSTR program.

job activity view

Report about address space consumption of resources. *See* view.

journal

Special-purpose data set that stores the chronological records of operator and system actions.

Journal log

Collection of messages. Journal logs are created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Journal log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Journal log stops when both data sets are filled and the first data set is not being processed by the archive program.

The TS Journal log is a single data set that wraps around when full.

line command

Command that you type in the line command column in a view or display. Line commands initiate actions that apply to the data displayed in that particular row.

line command column

Command input column on the left side of a view or display. *Contrast with* COMMAND line.

Log Edit

In the MAINVIEW for IMS Offline program named IMFLEDIT, function that extracts transaction (X‘FA’) and program (X‘F9’) records from the IMS system log. IMFLEDIT also extracts certain records that were recorded on the system log by IMS. IMFLEDIT then formats the records into a file called the IMS Resource Utilization File (IRUF).

MAINVIEW

BMC Software integrated systems management architecture.

MAINVIEW Alarm Manager (MV ALARM)

In conjunction with other MAINVIEW products, notifies you when an exception occurs. MAINVIEW Alarm Manager is capable of monitoring multiple systems simultaneously, which means that MAINVIEW Alarm Manager installed on one system keeps track of your entire sysplex. You can then display a single view that shows exceptions for all MAINVIEW performance monitors within your OS/390 or z/OS enterprise.

MAINVIEW Alternate Access

Enables MAINVIEW products to be used without TSO by providing access through EXCP and VTAM interfaces.

MAINVIEW Application Program Interface (MVAPI)

A CLIST- or REXX-based, callable interface that allows MAINVIEW AutoOPERATOR EXECs to access MAINVIEW monitor product view data.

MAINVIEW AutoOPERATOR

Product that uses tools, techniques, and facilities to automate routine operator tasks and provide online performance monitoring, and that achieves high availability through error minimization, improved productivity, and problem prediction and prevention.

MAINVIEW control area

In the MAINVIEW window environment, first three lines at the top of the view containing the window information line and the COMMAND, SCROLL, CURR WIN, and ALT WIN lines. The control area cannot be customized and is part of the information display. *Contrast with* MAINVIEW display area, MAINVIEW window area.

MAINVIEW Desktop Version of the MAINVIEW window interface designed to run on OS/2 and Windows workstations.

MAINVIEW display area

See MAINVIEW window area.

MAINVIEW Explorer Product that provides access to MAINVIEW products from a Web browser running under Windows. MAINVIEW Explorer replaces MAINVIEW Desktop.

MAINVIEW for CICS Product (formerly MV MANAGER for CICS) that provides real-time application performance analysis and monitoring for CICS system management.

MAINVIEW for DB2 Product (formerly MV MANAGER for DB2) that provides real-time and historical application performance analysis and monitoring for DB2 subsystem management.

MAINVIEW for DBCTL (MVDBC)

Product that provides real-time application performance analysis and monitoring for DBCTL management.

MAINVIEW for IMS (MVIMS) Offline

Product with a Performance Reporter component that organizes data and prints reports used to analyze IMS performance and a Transaction Accountant component that produces cost accounting and user charge-back records and reports.

MAINVIEW for IMS (MVIMS) Online

Product that provides real-time application performance analysis and monitoring for IMS management.

MAINVIEW for IP

Product that monitors OS/390 and z/OS mission-critical application performance as it relates to TCP/IP stack usage. Collected data includes availability, connections, response times, routers, service levels, storage, traffic, Web cache, and so on.

MAINVIEW for Linux–Servers

Product that allows you to monitor the performance of your Linux systems from the MAINVIEW windows interface.

MAINVIEW for MQSeries (formerly known as Command MQ for S/390)

Delivers comprehensive capabilities for configuration, administration, performance monitoring and operations management for an entire MQM (message queue manager) network.

MAINVIEW for OS/390

System management application (formerly known as MAINVIEW for MVS prior to version 2.5). Built upon the MAINVIEW window environment architecture, it uses the window interface to provide access to system performance data and other functions necessary in the overall management of an enterprise.

MAINVIEW for UNIX System Services

System management application that allows you to monitor the performance of the Unix System Services from a MAINVIEW window interface.

MAINVIEW for VTAM

Product that displays application performance data by application, transaction ID, and LU name. This collected data includes connections, response time statistics, application availability, and application throughput.

MAINVIEW for WebSphere Application Server (formerly known as MAINVIEW for WebSphere)

Product that provides extensive monitoring for the IBM WebSphere Application Server for z/OS and OS/390 environment.

MAINVIEW Selection Menu

ISPF selection panel that provides access to all MAINVIEW windows-mode and full-screen mode products.

MAINVIEW SRM

See MAINVIEW Storage Resource Manager (SRM).

MAINVIEW SRM DMS2HSM

Product that facilitates the conversion of CA-Disk, formerly known as DMS, to HSM.

MAINVIEW SRM EasyHSM

Product that provides online monitoring and reporting to help storage managers use DFHSM efficiently.

MAINVIEW SRM EasyPOOL

Product that provides control over data set allocation and enforcement of allocation and naming standards. EasyPOOL functions operate at the operating system level to intercept normal job processing, thus providing services without any JCL changes.

MAINVIEW SRM EasySMS

Product that provides tools that aid in the conversion to DFSMS and provides enhancement to the DFSMS environment after implementation. EasySMS consists of the EasyACS functions, the SMSACSTE function, and the Monitoring and Positioning Facility.

MAINVIEW SRM Enterprise Storage Automation

Product that delivers powerful event generation and storage automation technology across the storage enterprise. Used in conjunction with MAINVIEW AutoOPERATOR, automated solutions to perform pool, volume, application, or data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests.

MAINVIEW SRM SG-Auto

Product that provides early warning notification of storage anomalies and automated responses to those anomalies based on conditions in the storage subsystem.

MAINVIEW SRM SG-Control

Product that provides real-time monitoring, budgeting, and control of DASD space utilization.

MAINVIEW SRM StopX37/II

Product that provides enhancements to OS/390 or z/OS space management, reducing the incidence of space-related processing problems. The StopX37/II functions operate at the system level to intercept abend conditions or standards violations, thus providing services without any JCL changes.

MAINVIEW SRM StorageGUARD

Product that monitors and reports on DASD consumption and provides historical views to help control current and future DASD usage.

MAINVIEW Storage Resource Manager (SRM)

Suite of products that assist in all phases of OS/390 or z/OS storage management. MAINVIEW SRM consists of products that perform automation, reporting, trend analysis, and error correction for storage management.

MAINVIEW SYSPROG Services
See SYSPROG services.

MAINVIEW VistaPoint
Product that provides enterprise-wide views of performance. Application and workload views are available for CICS, DB2, DBCTL, IMS, OS/390, or z/OS. Data is summarized at the level of detail needed; for example, views can be for a single target, an OS/390 or z/OS image, or an entire enterprise.

MAINVIEW window area
Portion of the information display that is not the control area and in which views are displayed and windows opened. It includes all but the first three lines of the information display. *Contrast with* MAINVIEW control area.

monitor Online service that measures resources or workloads at user-defined intervals and issues warnings when user-defined thresholds are exceeded.

Multi-Level Automation (MLA)
The user-defined, multiple step process in Enterprise Storage Automation that implements solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved.

MVALARM *See* MAINVIEW Alarm Manager.

MVAPI *See* MAINVIEW Application Program Interface.

MVCICS *See* MAINVIEW for CICS.

MVDB2 *See* MAINVIEW for DB2.

MVDBC *See* MAINVIEW for DBCTL.

MVIMS *See* MAINVIEW for IMS.

MVIP *See* MAINVIEW for IP.

MVLNX *See* MAINVIEW for Linux–Servers.

MVMQ *See* MAINVIEW for MQSeries.

MVMVS *See* MAINVIEW for OS/390.

MVScope MAINVIEW for OS/390 application that traces both CPU usage down to the CSECT level and I/O usage down to the channel program level.

MVSRM *See* MAINVIEW Storage Resource Manager (SRM).

MVSRMHSM *See* MAINVIEW SRM EasyHSM.

MVSRMSGC *See* MAINVIEW SRM SG-Control.

MVSRMSGD *See* MAINVIEW SRM StorageGUARD.

MVSRMSGP *See* MAINVIEW SRM StorageGUARD.

MVUSS *See* MAINVIEW for UNIX System Services.

MVVP *See* MAINVIEW VistaPoint.

MVVTAM *See* MAINVIEW for VTAM.

MVWEB *See* MAINVIEW for WebSphere Application Server.

nested help Multiple layers of help pop-up windows. Each successive layer is accessed by clicking a hyperlink from the previous layer.

object Anything you can manipulate as a single unit. MAINVIEW objects can be any of the following: product, secondary window, view, row, column, or field.

You can issue an action against an object by issuing a line command in the line command column to the left of the object. *See* action.

OMVS workload Workload consisting of OS/390 OpenEdition address spaces.

online help Help information that is accessible online.

OS/390 and z/OS Installer
BMC Software common installation system for mainframe products.

OS/390 product address space (PAS)
Address space containing OS/390 or z/OS data collectors, including the CMF MONITOR Extractor. Used by MAINVIEW for OS/390, MAINVIEW for UNIX System Services, and CMF MONITOR products. *See* PAS.

parameter library Data set consisting of members that contain parameters for specific MAINVIEW products or a support component There can be several versions:

- the distributed parameter library, called BBPARM
- a site-specific parameter library or libraries

These can be

- a library created by AutoCustomization, called UBBPARM
- a library created manually, with a unique name

PAS

Product address space. Used by the MAINVIEW products. Contains data collectors and other product functions. *See also* OS/390 product address space (PAS) *and* BBI subsystem product address space (BBI-SS PAS).

performance group workload

Collection of address spaces defined to OS/390 or z/OS. If you are running OS/390 or z/OS with WLM in compatibility mode, MAINVIEW for OS/390 creates a performance group workload instead of a service class.

PERFORMANCE MANAGER

MAINVIEW for CICS online service for monitoring and managing current performance of CICS regions.

Performance Reporter (MVIMS)

MVIMS Offline component that organizes data and prints reports that can be used to analyze IMS performance.

Performance Reporter

Product component that generates offline batch reports. The following products can generate these reports:

- MAINVIEW for DB2
- MAINVIEW for CICS

Plex Manager

Product through which cross-system communication, MAINVIEW security, and an SSI context are established and controlled. Plex Manager is shipped with MAINVIEW window environment products as part of the coordinating address space (CAS) and is accessible as a menu option from the MAINVIEW Selection Menu.

pop-up display

Full-screen panel that displays additional information about a selected event in a detail trace.

pop-up window

Window containing help information that, when active, overlays part of the window area. A pop-up window is displayed when you issue the HELP command while working in windows-mode.

PRGP workload

In MVS/SP 5.0 or earlier, or in compatibility mode in MVS/SP 5.1 or later, composite of service classes. MAINVIEW for OS/390 creates a performance group workload for each performance group defined in the current IEAIPS*xx* member.

procedure library Data set consisting of members that contain executable procedures used by MAINVIEW AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. There can be several versions:

- the distributed parameter library, called BBPROC
- a site-specific parameter library or libraries

These can be

- a library created by AutoCustomization, called UBBPROC
- a library created manually, with a unique name

The site-created EXECs can be either user-written or customized MAINVIEW AutoOPERATOR-supplied EXECs from BBPROC.

product address space

See PAS.

profile library Data set consisting of members that contain profile information and cycle refresh definitions for a terminal session connected to a BBI-SS PAS. Other members are dynamically created by MAINVIEW applications. There can be several versions:

- the distributed profile library, called BBPROF
- a site-specific profile library or libraries

These can be

- a library created by AutoCustomization, called SBBPROF
- a library created manually, with a unique name

The site library is a common profile shared by all site users. The terminal session CLIST creates a user profile automatically if one does not exist; it is called userid.BBPROF, where userid is your logon ID. User profile libraries allow each user to specify unique PF keys, CYCLE commands, target system defaults, a Primary Option Menu, and a unique set of application profiles.

query One of two constituent parts of a view; the other is form. A query defines the data for a view; a form defines the display format. *See also* form, view.

real-time data Performance data as it exists at the moment of inquiry. Real-time data is recorded during the smallest unit of time for data collection. *Contrast with* historical data. *See also* current data and interval data.

Resource Analyzer Online real-time displays used to analyze IMS resources and determine which are affected by specific workload problems.

Resource Monitor Online data collection services used to monitor IMS resources and issue warnings when defined utilization thresholds are exceeded.

row (1) Horizontal component of a view or display comprising all the fields pertaining to a single device, address space, user, and so on. (2) Horizontal component of a DB2 table consisting of a sequence of values, one for each column of the table.

RxD2 Product that provides access to DB2 from REXX. It provides tools to query the DB2 catalog, issue dynamic SQL, test DB2 applications, analyze EXPLAIN data, generate DDL or DB2 utility JCL, edit DB2 table spaces, perform security administration, and much more.

sample cycle Time between data samples.

For the CMF MONITOR Extractor, this is the time specified in the extractor control statements (usually 1 to 5 seconds).

For real-time data, the cycle is not fixed. Data is sampled each time you press **Enter**.

sample library Data set consisting of members each of which contains one of the following items:

- sample JCL that can be edited to perform specific functions
- macro that is referenced in the assembly of user-written services
- sample user exit routine

There can be several versions:

- the distributed sample library, called BBSAMP
- a site-specific sample library or libraries

These can be

- a library created by AutoCustomization, called UBBSAMP
- a library created manually, with a unique name

sampler Program that monitors a specific aspect of system performance. Includes utilization thresholds used by the Exception Monitor. The CMF MONITOR Extractor contains samplers.

SBBPROF *See* profile library.

scope Subset of an SSI context. The scope could be all the data for the context or a subset of data within the context. It is user- or site-defined. *See* SSI context, target.

screen definition Configuration of one or more views that have been stored with the SAVEScr command and assigned a unique name. A screen includes the layout of the windows and the view, context, system, and product active in each window.

selection view In MAINVIEW products, view displaying a list of available views.

service class workload

Collection of address spaces defined to OS/390 or z/OS. If you are running Workload Manager (WLM) in goal mode, MAINVIEW for OS/390 creates a service class workload for each service class that you define through WLM definition dialogs.

If you are running MVS 4.3 or earlier, or MVS/SP 5.1 or later with WLM in compatibility mode, OS/390 creates a performance group workload instead of a service class. *See* performance group workload.

service objective Workload performance goal, specified in terms of response time for TSO workloads or turnaround time for batch workloads. Performance group workloads can be measured by either objective. Composite workload service objectives consist of user-defined weighting factors assigned to each constituent workload. For compatibility mode, neither OS/390 nor z/OS provides any way to measure service.

service point Specification, to MAINVIEW, of the services required to enable a specific product. Services can be actions, selectors, or views. Each target (for example, CICS, DB2, or IMS) has its own service point.

The PLEX view lists all the defined service points known to the CAS to which the terminal session is connected.

service request block (SRB)

Control block that represents a routine to be dispatched. SRB mode routines generally perform work for the operating system at a high priority. An SRB is similar to a task control block (TCB) in that it identifies a unit of work to the system. *See also* task control block.

service select code Code entered to invoke analyzers, monitors, and general services. This code is also the name of the individual service.

session Total period of time an address space has been active. A session begins when monitoring can be performed. If the product address space (PAS) starts after the job, the session starts with the PAS.

SG-Auto *See* MAINVIEW SRM SG-Auto.

SG-Control *See* MAINVIEW SRM SG-Control.

single system image (SSI)

Feature of the MAINVIEW window environment architecture where you can view and perform actions on multiple OS/390or z/OS systems as though they were a single system. The rows of a single tabular view can contain rows from different OS/390 or z/OS images.

Skeleton Tailoring Facility

A facility in MAINVIEW AutoOPERATOR that allows skeleton JCL to be used during job submission. Skeleton JCL can contain variables within the JCL statements to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility functions similar to the TSO skeleton tailoring facility.

SRB

See service request block.

SSI

See single system image.

SSI context

Name created to represent one or more targets for a given product. *See* context, target.

started task workload

Address spaces running jobs that were initiated programmatically.

statistics interval

For MAINVIEW for DB2, cumulative count within a predefined interval (30-minute default set by the DB2STATS parameter in the distributed BBPARM member BBIISP00) for an analyzer service DELTA or RATE display. Specifying the DELTA parameter displays the current value as the difference between the value sampled by the current analyzer request and the value sampled at the start of the current interval. Specifying the RATE parameter displays the current value by minute (DELTA divided by the number of elapsed minutes).

stem variables

A REXX facility, supported in MAINVIEW AutoOPERATOR REXX EXECs and the Skeleton Tailoring Facility, where variable names end with a period followed by a number, such as &POOL.1. This configuration allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array. For example, &POOL.0 = 5 would indicate variables &POOL.1 through &POOL.5 exist.

StopX37/II

See MAINVIEW SRM StopX37/II.

StorageGUARD

See MAINVIEW SRM StorageGUARD.

summary view

View created from a tabular view using the Summarize option in view customization. A summary view compresses several rows of data into a single row based on the summarize criteria.

SYSPROG services Component of MAINVIEW for OS/390. Over 100 services that detect, diagnose, and correct OS/390 or z/OS system problems as they occur. Accessible from the OS/390 Performance and Control Main Menu. Note that this component is also available as a stand-alone product MAINVIEW SYSPROG Services.

system resource *See* object.

target Entity monitored by one or more MAINVIEW products, such as an OS/390 or z/OS image, an IMS or DB2 subsystem, a CICS region, or related workloads across systems. *See* context, scope, SSI context.

target context Single target/product combination. *See* context.

TASCOSTR MAINVIEW for IMS Offline program that summarizes detail and summary IMS Resource Utilization Files (IRUFs) to be used as input to the offline components.

task control block (TCB)
Address space-specific control block that represents a unit of work that is dispatched in the address space in which it was created. *See also* service request block.

TCB *See* task control block.

terminal session (TS)
Single point of control for MAINVIEW products, allowing data manipulation and data display and providing other terminal user services for MAINVIEW products. The terminal session runs in a user address space (either a TSO address space or a stand-alone address space for EXCP/VTAM access).

TDIR *See* trace log directory.

threshold Specified value used to determine whether the data in a field meets specific criteria.

TLDS *See* trace log data set.

total mode Usage mode in CMFMON wherein certain columns of data reflect the cumulative value between collection intervals. Invoked by the DELta OFF command. *See also* collection interval, delta mode.

trace (1) Record of a series of events chronologically listed as they occur. (2) Online data collection and display services that track transaction activity through DB2, IMS, or CICS.

trace log data set (TLDS)

Single or multiple external VSAM data sets containing summary or detail trace data for later viewing or printing. The trace log(s) can be defined as needed or dynamically allocated by the BBI-SS PAS. Each trace request is assigned its own trace log data set(s).

trace log directory (TDIR)

VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time of data set creation, the current status of the data set, the trace target, and other related information.

transaction Specific set of input data that initiates a predefined process or job.

Transaction Accountant

MVIMS Offline component that produces cost accounting and user charge-back records and reports.

TS *See* terminal session.

TSO workload Workload that consists of address spaces running TSO sessions.

UAS *See* user address space.

UBBPARM *See* parameter library.

UBBPROC *See* procedure library.

UBBSAMP *See* sample library.

user address space

Runs a MAINVIEW terminal session (TS) in TSO, VTAM, or EXCP mode.

User BBPROF *See* profile library.

view Formatted data within a MAINVIEW window, acquired from a product as a result of a view command or action. A view consists of two parts: query and form. *See also* form, job activity view, query.

view definition Meaning of data that appears online, including source of data, selection criteria for data field inclusion and placement, data format, summarization, context, product, view name, hyperlink fields, and threshold conditions.

view command Name of a view that you type on the COMMAND line to display that view.

view command stack

Internal stack of up to 10 queries. For each command, the stack contains the filter parameters, sort order, context, product, and time frame that accompany the view.

view help Online help describing the purpose of a view. To display view help, place the cursor on the view name on the window information line and press **PF1** (HELP).

window Area of the MAINVIEW screen in which views and resources are presented. A window has visible boundaries and can be smaller than or equal in size to the MAINVIEW window area. *See* active window, alternate window, current window, MAINVIEW window area.

window information line

Top border of a window. Shows the window identifier, the name of the view displayed in the window, the system, the scope, the product reflected by the window, and the tomfooleries for which the data in the window is relevant. *See also* window status field.

window number Sequential number assigned by MAINVIEW to each window when it is opened. The window number is the second character in the window status field. *See also* window status field.

window status One-character letter in the window status field that indicates when a window is ready to receive commands, is busy processing commands, is not to be updated, or contains no data. It also indicates when an error has occurred in a window. The window status is the first character in the window status field. *See also* window information line, window status field.

window status field Field on the window information line that shows the current status and assigned number of the window. *See also* window number, window status.

windows mode Display of one or more MAINVIEW product views on a screen that can be divided into a maximum of 20 windows. A window information line defines the top border of each window. *Contrast with* full-screen mode.

WLM workload In goal mode in MVS/SP 5.1 and later, a composite of service classes. MAINVIEW for OS/390 creates a workload for each WLM workload defined in the active service policy.

workflow Measure of system activity that indicates how efficiently system resources are serving the jobs in a workload.

workload (1) Systematic grouping of units of work (for example, address spaces, CICS transactions, IMS transactions) according to classification criteria established by a system administrator. (2) In OS/390 or z/OS, a group of service classes within a service definition.

workload activity view

Tracks workload activity as the workload accesses system resources. A workload activity view measures workload activity in terms of resource consumption and how well the workload activity meets its service objectives.

Workload Analyzer Online data collection and display services used to analyze IMS workloads and determine problem causes.

workload definition Workload created through the WKLIST view. Contains a unique name, a description, an initial status, a current status, and selection criteria by which address spaces are selected for inclusion in the workload. *See* Workload Definition Facility.

Workload Definition Facility

In MAINVIEW for OS/390, WKLIST view and its associated dialogs through which workloads are defined and service objectives set.

workload delay view

Tracks workload performance as the workload accesses system resources. A workload delay view measures any delay a workload experiences as it contends for those resources.

Workload Monitor Online data collection services used to monitor IMS workloads and issue warnings when defined thresholds are exceeded.

workload objectives

Performance goals for a workload, defined in WKLIST. Objectives can include measures of performance such as response times and batch turnaround times.

Index

STOP!

IMPORTANT INFORMATION - DO NOT INSTALL THIS PRODUCT UNLESS YOU HAVE READ ALL OF THE FOLLOWING MATERIAL

By clicking the YES or ACCEPT button below (when applicable), or by installing and using this Product or by having it installed and used on your behalf, You are taking affirmative action to signify that You are entering into a legal agreement and are agreeing to be bound by its terms, EVEN WITHOUT YOUR SIGNATURE. BMC is willing to license this Product to You ONLY if You are willing to accept all of these terms. CAREFULLY READ THIS AGREEMENT. If You DO NOT AGREE with its terms, DO NOT install or use this Product; press the NO or REJECT button below (when applicable) or promptly contact BMC or your BMC reseller and your money will be refunded if by such time You have already purchased a full-use License.

SOFTWARE LICENSE AGREEMENT FOR BMC PRODUCTS

SCOPE. This is a legally binding Software License Agreement (**"License"**) between You (either an individual or an entity) and BMC pertaining to the original computer files (including all computer programs and data stored in such files) contained in the enclosed Media (as defined below) or made accessible to You for electronic delivery, if as a prerequisite to such accessibility You are required to indicate your acceptance of the terms of this License, and all whole or partial copies thereof, including modified copies and portions merged into other programs (collectively, the **"Software"**). **"Documentation"** means the related hard-copy or electronically reproducible technical documents furnished in association with the Software, **"Media"** means the original BMC-supplied physical materials (if any) containing the Software and/or Documentation, **"Product"** means collectively the Media, Software, and Documentation, and all Product updates subsequently provided to You, and **"You"** means the owner or lessee of the hardware on which the Software is installed and/or used. **"BMC"** means BMC Software Distribution, Inc. unless You are located in one of the following regions, in which case "BMC" refers to the following indicated BMC Software, Inc. subsidiary: (i) Europe, Middle East or Africa --BMC Software Distribution, B.V., (ii) Asia/Pacific -- BMC Software Asia Pacific Pte Ltd., (iii) Brazil -- BMC Software do Brazil, or (iv) Japan -- BMC Software K.K. **If You enter into a separate, written software license agreement signed by both You and BMC or your authorized BMC reseller granting to you the rights to install and use this Product, then the terms of that separate, signed agreement will apply and this License is void.**

FULL-USE LICENSE. Subject to these terms and payment of the applicable license fees, BMC grants You this non-exclusive License to install and use one copy of the Software for your internal use on the number(s) and type(s) of servers or workstations for which You have paid or agreed to pay to BMC or your BMC reseller the appropriate license fee. If your license fee entitles You only to a License having a limited term, then the duration of this License is limited to that term; otherwise this License is perpetual, subject to the termination provisions below.

TRIAL LICENSE. If You have not paid or agreed to pay to BMC or your BMC Reseller the appropriate license fees for a full use license, then, **NOTWITHSTANDING ANYTHING TO THE CONTRARY CONTAINED IN THIS LICENSE**: (i) this License consists of a non-exclusive evaluation license ("Trial License") to use the Product for a limited time ("Trial Period") only for evaluation; (ii) during the Trial Period, You may not use the Software for development, commercial, production, database management or other purposes than those expressly permitted in clause (i) immediately above; and (iii) your use of the Product is on an ***AS IS*** basis, and **BMC, ITS RESELLERS AND LICENSORS GRANT NO WARRANTIES OR CONDITIONS (INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) TO YOU AND ACCEPT NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE.** If You use this Product for other than evaluation purposes or wish to continue using it after the Trial Period, you must purchase a full-use license. When the Trial Period ends, your right to use this Product automatically expires, though in certain cases You may be able to extend the term of the Trial Period by request. Contact BMC or your BMC reseller for details.

TERM AND TERMINATION. This License takes effect on the first to occur of the date of shipment or accessibility to You for electronic delivery, as applicable (the **"Product Effective Date"**). You may terminate this License at any time for any reason by written notice to BMC or your BMC reseller. This License and your right to use the Product will terminate automatically with or without notice by BMC if You fail to comply with any material term of this License. Upon termination, You must erase or destroy all components of the Product including all copies of the Software, and stop using or accessing the Software. Provisions concerning Title and Copyright, Restrictions (or Restricted Rights, if You are a U.S. Government entity) or limiting BMC's liability or responsibility shall survive any such termination.

TITLE AND COPYRIGHT; RESTRICTIONS. All title and copyrights in and to the Product, including but not limited to all modifications thereto, are owned by BMC and/or its affiliates and licensors, and are protected by both United States copyright law and applicable international copyright treaties. You will not claim or assert title to or ownership of the Product. To the extent expressly permitted by applicable law or treaty notwithstanding this limitation, You may copy the Software only for backup or archival purposes, or as an essential step in utilizing the Software, but for no other purpose. You will not remove or alter any copyright or proprietary notice from copies of the Product. You acknowledge that the Product contains valuable trade secrets of BMC and/or its affiliates and licensors. Except in accordance with the terms of this License, You agree (a) not to decompile, disassemble, reverse engineer or otherwise attempt to derive the Software's source code from object code except to the extent expressly permitted by applicable law or treaty despite this limitation; (b) not to sell, rent, lease, license, sublicense, display, modify, time share, outsource or otherwise transfer the Product to, or permit the use of this Product by, any third party; and (c) to use reasonable care and protection to prevent the unauthorized use, copying, publication or dissemination of the Product and BMC confidential information learned from your use of the Product. **You will not export or re-export any Product without both the written consent of BMC and the appropriate U.S. and/ or foreign government license(s) or license exception(s).** Any programs, utilities, modules or other software or documentation created, developed, modified or enhanced by or for You using this Product shall likewise be subject to these restrictions. BMC has the right to obtain injunctive relief against any actual or threatened violation of these restrictions, in addition to any other available remedies. Additional restrictions may apply to certain files, programs or data supplied by third parties and embedded in the Product; consult the Product installation instructions or Release Notes for details.

LIMITED WARRANTY AND CONDITION. If You have purchased a Full-Use License, BMC warrants that (i) the Media will be, under normal use, free from physical defects, and (ii) for a period of ninety (90) days from the Product Effective Date, the Product will perform in substantial accordance with the operating specifications contained in the Documentation that is most current at the Product Effective Date. BMC's entire liability and your exclusive remedy under this provision will be for BMC to use reasonable best efforts to remedy defects covered by this warranty and condition within a reasonable period of time or, at BMC's option, either to replace the defective Product or to refund the amount paid by You to license the use of the Product. BMC and its suppliers do not warrant that the Product will satisfy your requirements, that the operation of the Product will be uninterrupted or error free, or that all software defects can be corrected. This warranty and condition shall not apply if: (i) the Product is not used in accordance with BMC's instructions, (ii) a Product defect has been caused by any of your or a third party's malfunctioning equipment, (iii) any other cause within your control causes the Product to malfunction, or (iv) You have made modifications to the Product not expressly authorized in writing by BMC. No employee, agent or representative of BMC has authority to bind BMC to any oral representations, warranties or conditions concerning the Product. **THIS WARRANTY AND CONDITION IS IN LIEU OF ALL OTHER WARRANTIES AND CONDITIONS. THERE ARE NO OTHER EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS LICENSE OR ANY PRODUCT LICENSED HEREUNDER. THIS PARAGRAPH SHALL NOT APPLY TO A TRIAL LICENSE.** Additional support and maintenance may be available for an additional charge; contact BMC or your BMC reseller for details.

LIMITATION OF LIABILITY. Except as stated in the next succeeding paragraph, BMC's and your BMC reseller's total liability for all damages in connection with this License is limited to the price paid for the License. **IN NO EVENT SHALL BMC BE LIABLE FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE USE OF THIS PRODUCT (SUCH AS LOSS OF PROFITS, GOODWILL, BUSINESS, DATA OR COMPUTER TIME, OR THE COSTS OF RECREATING LOST DATA), EVEN IF BMC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Some jurisdictions do not permit the limitation of consequential damages so the above limitation may not apply.

INDEMNIFICATION FOR INFRINGEMENT. BMC will defend or settle, at its own expense, any claim against You by a third party asserting that your use of the Product within the scope of this License violates such third party's patent, copyright, trademark, trade secret or other proprietary rights, and will indemnify You against any damages finally awarded against You arising out of such claim. However, You must promptly notify BMC in writing after first receiving notice of any such claim, and BMC will have sole control of the defense of any action and all negotiations for its settlement or compromise, with your reasonable assistance. BMC will not be liable for any costs or expenditures incurred by You without BMC's prior written consent. If an order is obtained against your use of the Product by reason of any claimed infringement, or if in BMC's opinion the Product is likely to become the subject of such a claim, BMC will at its option and expense either (i) procure for You the right to continue using the product, or (ii) modify or replace the Product with a compatible, functionally equivalent, non-infringing Product, or (iii) if neither (i) nor (ii) is practicable, issue to You a pro-rata refund of your paid license fee(s) proportionate to the number of months remaining in the 36 month period following the Product Effective Date. This paragraph sets forth your only remedies and the total liability to You of BMC, its resellers and licensors arising out of such claims.

GENERAL. This License is the entire understanding between You and BMC concerning this License and may be modified only in a mutually signed writing between You and BMC. If any part of it is invalid or unenforceable, that part will be construed, limited, modified, or, severed so as to eliminate its invalidity or unenforceability. This License will be governed by and interpreted under the laws of the jurisdiction named below, without regard to conflicts of law principles, depending on which BMC Software, Inc. subsidiary is the party to this License: (i) BMC Software Distribution, Inc. - the State of Texas, U.S.A., (ii) BMC Software Distribution, B.V. - The Netherlands, (iii) BMC Software Asia Pacific Pte Ltd. -- Singapore (iv) BMC Software do Brazil -- Brazil, or (v) BMC Software K.K. -- Japan. Any person who accepts or signs changes to the terms of this License promises that they have read and understood these terms, that they have the authority to accept on your behalf and legally obligate You to this License. Under local law and treaties, the restrictions and limitations of this License may not apply to You; You may have other rights and remedies, and be subject to other restrictions and limitations.

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

Notes